CANADIAN THESES ON MICROFICHE

THÈSES CANADIENNES SUR MICROFICHE

National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Direction du développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

NL 339 (r. 85/09)

Canada

0-315-19582-9

National Library of Canada
Canadian Theses Division
Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Division des thèses canadiennes

159

67410

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

- Please print or type — Écrire en lettres moulées ou dactylographier

Full Name of Author — Nom complet de l'auteur
Roessler, Jochen

Date of Birth — Date de naissance 27.06.49.	Country of Birth — Lieu de naissance Germany

Permanent Address — Résidence fixe
Geogr. Dep./Met. Div., U/A T6G 2H4

Title of Thesis — Titre de la thèse
A Study of Turbulence using Bifurcation Methods

University — Université
Univ. of Alberta

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée
PhD

Year this degree conferred — Année d'obtention de ce grade 1984	Name of Supervisor — Nom du directeur de thèse E. R. Reinelt

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

Date 13.08.84.	Signature

NL-91 (4/77)

THE UNIVERSITY OF ALBERTA

A STUDY OF TURBULENCE USING BIFURCATION METHODS

by

© JOCHEN ROESSLER

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

IN

METEOROLOGY
DEPARTMENT OF GEOGRAPHY

EDMONTON, ALBERTA
FALL, 1984

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR Jochen Roessler

TITLE OF THESIS A Study of Turbulence using Bifurcation Methods

DEGREE FOR WHICH THESIS WAS PRESENTED Doctor of Philosophy

YEAR THIS DEGREE GRANTED 1984

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(Signed)

PERMANENT ADDRESS:

DATED 13. 08. 84

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled A STUDY OF TURBULENCE USING BIFURCATION METHODS submitted by JOCHEM ROESSLER in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

E R Reinelt

Supervisor

G. J. Bulle

K. D. Hage

Paul Arminjon

External Examiner

Date 13 July 1984

ABSTRACT

Until the late sixties and seventies, the main approaches to treating fluid dynamics in the turbulent regime were based on statistical techniques and semiempirical models. The advancements in applied and nonlinear mathematics initiated research in the stability properties of equations governing fluid motion. This allowed the modelling of the breakdown of certain flow structures into the chaotic state. Physical models describing possible mechanisms of turbulence in connection with the aforementioned stability properties of the governing equations are still rare. The present work describes such a model and its machine simulation.

The mechanism for the generation of turbulence assumes a sequence of three Hopf-bifurcations on the Navier-Stokes equations. This leads to a 3-torus as invariant manifold in state space. The vector fields on this 3-torus are structurally unstable and chaotic behaviour is triggered by combinations of small random perturbations. To simulate this mechanism, two Navier-Stokes type equations for the evoluation of time dependent velocity perturbations are derived and discretized in three dimensions by using finite elements and the Galerkin method. The resulting systems of ordinary differential equations are then tested by a FORTRAN program for Hopf bifurcations and solution curves in state space were obtained numerically on the computer. Small random perturbations are imposed on these solution curves by using random number generators and a spectral analysis for

energy densities using Fast Fourier transform under use of the Taylor hypothesis is performed on several of these solutions with and without random perturbation in order to obtain energy spectra which permit comparison with the inertial subrange law for homogeneous and isotropic turbulence.

ACKNOWLEDGEMENTS

The study here presented was possible due to the advice and support of several people. However, the guidance and feedback of some was invaluable. I, therefore, like to thank Dr. J. Timourian for helping me to obtain an understanding for differential topology which is vital for the development of the theory and mechanism of the presented model.

In view of the fact that my experience and knowledge of FORTRAN was next to nonexistent at the beginning, the development of a discretization program with three-dimensional finite elements would not have been possible for me without relying on the expertise and guidance of Dr. J. Honsaker and D. Oracheski. Likewise, I am indebted to Dr. B. Hassard from SUNY, Buffalo for providing me with his program BIFOR2 and to W.G. Aiello for implementing this program at the U of A computer as well as providing me with about half of the total CPU time used for this study.

My thanks go also to D. Thornton and Dr. R. Torgerson for giving me access to the array processor at Computing Services and helping me to implement and understand the performance of some of the equation solver programs on this device as well as helping out with some extra CPU time.

For fruitful discussions and relevant feedback I owe thanks to Dr. R.B. Charlton and Dr. K.D. Hage as well as to my external examiner Dr. P. Arminjon for investing a considerable amount of time in evaluating my research.

An outstanding job on the nontrivial graphics work required was done by M. Fisher and S. Kucharyshyn which I am therefore indebted to as well as to V. Spak for expert typing of the manuscript and skillful retyping and rearranging to fit a myriad of changes and corrections.

Last, but by no means least, it is my pleasure to express thanks to my supervisor Dr. E. Reinelt for securing far more than the average amount of available CPU time for my research and for facilitating my financial survival by providing a part time job and to my cosupervisor Dr. G. Butler for adopting me as an extra burden on his already overloaded timetable. Both of them demonstrated a high degree of confidence in me by giving me the go ahead for a rather unorthodox approach towards the problem under study.

TABLE OF CONTENTS

PAGE

INTRODUCTION 1

PART 1 6

CHAPTER 1. THE HOPF BIFURCATION 6

CHAPTER 2. THE WORK OF RUELLE AND TAKENS 7

CHAPTER 3. THE MODEL AND ITS MECHANISM 18

Informal Description 18

Mathematical Formulation 25

PART 2 35

REMARKS ON PART 2: Technical Realization and Approximation of the Model 35

CHAPTER 4. THE NAVIER-STOKES EQUATIONS FOR PERTURBATIONS . . . 35

Discretization of the NS-equation 38

CHAPTER 5. THE GALERKIN METHOD 39

CHAPTER 6. THE DIVERGENCE-FREE BASIS FUNCTIONS 53

CHAPTER 7. DIRICHLET BOUNDARY CONDITIONS AND STRUCTURE OF MEAN FLOW 57

PART 3 62

CHAPTER 8. INVESTIGATION OF BIFURCATION PROPERTIES 62

CHAPTER 9. APPLICATION OF ODE SOLVERS 68

CHAPTER 10. SIMULATION OF RANDOM PERTURBATIONS AND THEIR EFFECT (ON FLOW IV) 74

CHAPTER 11. SPECTRAL ANALYSIS 76

CHAPTER 12. SUMMARY AND CONCLUSIONS 81

PAGE

REFERENCES 90

APPENDIX A 93

Definitions and Explanations of some Terms and Concepts Used in Part 1.

APPENDIX B 102

Derivation of some Technical Equations and Details Used in Part 2.

Description of the Program DSCRB.

APPENDIX C 130

Listing and Description of Flows Tested for Bifurcations.

APPENDIX D 135

Introduction to some Numerical Work Done on the Navier-Stokes Equation.

APPENDIX E 139

The Inertial Subrange Law: Kolmogorov's and Heisenberg's Derivation.

The Taylor Hypothesis.

APPENDIX F 151

Trajectory Projections and Spectral Graphs.

APPENDIX G 297

Program Listings.

LIST OF TABLES

PAGE

TABLE 1 80

LIST OF FIGURES

PAGE

FIGURE 1. 2-Torus with Trajectory and/or Vector Field on its Surface with Velocity Components $\vec{V}_1, \vec{V}_2$ and Periods p_1, p_2 9

FIGURE 2. Poincare Map on Plane Transversal to Hopf Orbit . . 11

FIGURE 3. Circle Oscillating within Annulus 11

FIGURE 4. 2-Torus Oscillating within Torus Shell 14

FIGURE 5. Horseshoe-Diffeomorphism: First Mapping 96

FIGURE 6. Horseshoe-Diffeomorphism: Second Mapping 96

FIGURE 7. Horseshoe Diffeomorphism Sketched on the Outer Surface of the Torus Shell 16

FIGURE 8. Vector Field on Oscillating 2-Torus with Normal Components for Contraction and Expansion 21

FIGURE 9. Vector Fields on 3-Torus Section with Overlapping Walls Region 22

FIGURE 10. Local Vector Field Structure of 2-Torus in 3-Dimensional Subspace 22

FIGURE 11. Two Tetrahedrons Sharing a Face with Centroid (Support for Finite Elements) 55

FIGURE 12A. A-Configuration of Discretization 58

FIGURE 12B. B-Configuration of Discretization 59

PAGE

FIGURE 13. Flow Through Domain D by $\vec{U}$ 61

FIGURE 14. Structure of Nonlinear Matrix W^{∇}_{onm} 127

FIGURE 15. Outer Loops of DSCRB 128

FIGURE 16. Inner Loops of DSCRB 129

FIGURE 17. Program Chart 301

INTRODUCTION

Perhaps one of the main reasons why the phenomenon of turbulence lacks any satisfactory description or model is the fact that no completely satisfactory definition of turbulence seems to be possible. One of the simplest and straightforward definitions might be 'a state of continuous instability' [Tritton, [1]]. This again can be understood only intuitively, because the concepts of continuity and instability are almost contradictory; continuous processes take place over a considerable length of time, while the phenomenon of instability occurs as a sudden breakdown of a state and the following transition into another one. Looking at turbulence, one cannot observe different states at different points in time, if the governing external parameters, such as temperature, temperature gradient, mean shear, or viscosity are not changed. Moreover, it is not certain whether there exists only one main mechanism in nature, which governs all the different types of observed turbulence.

So far, only methods of statistical analysis, which lead to the various techniques of finite closure of moment equations derived from the Navier-Stokes equations* have given results which are comparable with experiments. One of the most outstanding results in this respect is the inertial subrange law first derived by Kolmogorov using dimensional analysis on a rather heuristic model of energy transfer within isotropic turbulent flow. Using methods of statistical physics and finite closure techniques applied to these moment equations, Heisenberg reproduced this result which can be verified experimentally (see Appendix E). It is sometimes also referred to as the (-5/3)-law, since it states the proportionality between spectral energy density E and wave number k to the power of -5/3 :

---

*They will be referred to as NS-equations from now on.

$$E \sim k^{-5/3} .$$

For an introduction to this result see e.g. [1] or Appendix E.

The application of methods from differential topology and ergodic theory to the qualitative theory of differential equations* lead to the development of concepts such as structural stability and strange attractors, and thereby to a better understanding of their solution spaces. However, due to the difficulties encountered, only very slow progress can be made even for low-dimensional systems of ordinary differential equations.** Complete results about structural stability exist so far only for two-dimensional (2D) systems and the term 'strange attractor' was coined to reflect the lack of understanding of the phenomenon it tries to depict. This term is used mainly in connection with chaotic behaviour of solutions of differential equations and the attracting regions in solution space associated therewith. In the context of fluid dynamics chaotic behaviour is called turbulence; again one experiences the lack of ability to come to a satisfactory definition and has to resort to terms such as 'strange attractor'.

---

[*] For an introduction, see e.g.: Nemytskii and Stepanov: Qualitative theory of differential equation. Princeton.

[**] They will be referred to as ODE-systems from now on.

Since the set of NS-equations is far from being a low-dimensional ODE-system but rather a system of three nonlinear, parabolic partial differential equations*, its state space is infinite-dimensional. Moreover, its solution type can change drastically with the boundary conditions or the geometry of fluid domain imposed.

As a consequence, the analysis of equations of the NS-type is being conducted in two related, but nevertheless different ways. One approach, mainly taken by engineers, deals with very special boundary conditions and tries to find the associated flows as solutions of the NS-equations which, as a rule, first have to be adapted to the problem at hand. In this connection, various techniques of discretizing the fluid domain can be used to obtain numerical solutions. The other way is to approach the NS-equations on a very general level and try to analyze the structure of its solution space. This is mainly done by mathematicians, since the techniques involved require extensive use of higher analysis, differential topology and function theory.

One of the better known publications in this field which suggests a mathematical mechanism for the generation of turbulence and chaos originating from solutions of equations of the NS-type is the paper of Ruelle and Takens [2]. These authors have shown that the proposition of Landau and Lifschitz [3] for turbulent motion as a quasi-periodic function in time is invalid.

The present study was largely motivated by the aforementioned paper of Ruelle and Takens. Therefore, after an introduction to the Hopf bifurcation (Chapter 1), which is the main tool for the construction of the mathematical mechanism, a rather informal discussion of this paper follows (Chapter 2). After that, the model, which is the main subject of the present study, and its mechanism are explained (Chapter 3). This is done, first, by an informal description aimed at explaining the basic ideas in terms as simple as possible. Following this rather heuristic introduction is a mathematical formulation of the same model. The necessary function spaces are introduced, which permit representing of the model by means of equations. This concludes the first part and therewith the description of the theoretical basis.

---

*They will be referred to as PDE from now on.

Part 2 describes the technical realization of the ideas described in Part One. First, a set of equations of the NS-type is developed for a set of perturbations on the fluid velocity superimposed on each other (Chapter 4). To obtain numerical solutions for this set of equations, a discretization of the domain in spaces governed by these equations is carried out, based on finite element techniques and the Galerkin method (Chapter 5). This leads to an approximation of NS-type PDE's by a system of ODE's. In connection with this discretization, divergence-free basis functions or finite elements (Chapter 6) and the use of Dirichlet boundary condition for the domain of the velocity perturbations are introduced and a particular mean flow structure is discussed (Chapter 7).

Part 3 describes the numerical study of the system of ODE's obtained through the discretization discussed in Part 2. First, the system of ODE's is investigated for its bifurcation properties for different profiles of the underlying mean flow (Chapter 8). The results of four particular mean flows are selected for further analysis of their stability characteristics by use of numerical ODE-solvers and graphs of their solution trajectories (Chapter 9). One of these four flows is studied under the superposition of numerically simulated random perturbations (Chapter 10) and the results are submitted to a spectral analysis to investigate any realization of the inertial subrange law (Chapter 11).

Finally, the summary and conclusions (Chapter 12) contain an overall interpretation of the results obtained in the three parts. Also, a number of technical restrictions and their effect and, in connection therewith, the comparability of the spectral analysis with experimentally obtained spectra is considered.

PART 1

1. THE HOPF BIFURCATION

The following introduction to the Hopf bifurcation is rather informal and concise. Although Ruelle and Takens apply it to NS-type PDE's it is being introduced here for ODE's for the sake of simplicity. More elaborate descriptions with theorems and proofs can be found in various publications, see e.g., Marsden and McCracken [4] or Hassard, Kazarinoff and Wan [5] or Hopf's original paper [6].

Consider an n-dimensional ODE-system $(n \geq 2)$:

$$X'_\mu(t) = f(X_\mu(t),\mu) \tag{1.1}$$

For the real parameter μ within a certain interval I, the system is assumed to have the stationary solution $X_{\mu s}$ according to $X'_\mu(t) = 0$, i.e.,

$$0 = f(X_{\mu s},\mu), \quad \mu \in I. \tag{1.2}$$

Taking the variational derivative of (1.1) at X_{μ_L} yields the linearized system:

$$X'_\mu(t) = A(\mu_L)X_\mu(t); \quad A(\mu_L) \equiv \left| \frac{\partial f(X_\mu(t),\mu_L)}{\partial X_\mu(t)} \right|_{X_\mu(t)=X_{\mu_L}(t)} \tag{1.3}$$

$A(\mu_L)$ is the Jacobian matrix of the system at $X_\mu(t) = X_{\mu_L}(t)$. A Hopf bifurcation, i.e., the appearance of a periodic solution $X_{\mu p}(t)$ which grows in amplitude with increasing or decreasing μ out of the stationary solution at $\mu = \mu_c$, occurs when the following conditions are satisfied:

(I) A pair of complex conjugate eigenvalues $\lambda(\mu_c)$ and $\overline{\lambda}(\mu_c)$ of $A(\mu_c)$ cross the imaginary axis at $\mu = \mu_c$.

(II) All the remaining eigenvalues of $A(\mu_L)$ stay away from the imaginary axis for μ near μ_c.

(III) $\mathrm{Re}\,\lambda(\mu_c) = 0; \quad \mathrm{Im}\,\lambda(\mu_c) \neq 0$.

(IV) $\left.\frac{d\lambda(\mu)}{d\mu}\right|_{\mu=\mu_c} \neq 0$.

Properties of stability and criticality of the periodic solution, i.e., on what side of μ_c $X_{\mu p}(t)$ exists and whether it is stable or unstable require a more detailed analysis of the nonlinear terms in Equation (1.1).

2. THE WORK OF RUELLE AND TAKENS

In their paper Ruelle and Takens consider a sequence of Hopf bifurcations occuring to equations of the type (1.1). However, they do not restrict themselves to ODE's and include PDE's expecially of the NS-type. The parameter μ can therefore be, e.g., mean velocity, viscosity, shear or quantities like Reynolds number, Rayleigh number etc. depending on the type of application. Moreover, the right-hand side of Equation (1.1) has to be understood as an operator generating a vector field in state space* H of Equation (1.1). In the ODE case this 'RHS-operator' can be represented as a combination of matrices with not necessarily linear properties as far as its effect on elements in H is concerned. In the case of PDE's the combination consists of the standard differential operators such as gradient, divergence and Laplacean, again with possibly nonlinear behaviour. The eigenvalues to be considered are now the ones of the linearized forms of these 'RHS-operators'.

---

*This space is also called phase space by some authors, see e.g. [9].

The first part of the paper contains a description of a potential mechanism for turbulence by successively advancing pairs of complex-conjugate eigenvalues of the linearized RHS-operators over the imaginary axis as the critical parameter μ increases. As the first pair crosses the imaginary axis, a fixed point in solution space (i.e. a stable stationary solution) exchanges stability with a family of closed Hopf orbits (i.e., periodic solutions) which grow monotonically with μ. The crossing of the second pair causes this first orbit to loose stability and degenerate into a trajectory on the surface of a 2-torus. This 2-torus is an attracting invariant manifold, i.e. the trajectory of any solution will eventually end up on its surface. Further increase of the parameter μ causes the third pair of complex-conjugate eigenvalues to cross the imaginary axis and the degeneration of the 2-torus into a 3-torus. This is the point where the subject matter becomes difficult to understand, since a 3-torus can only 'exist' in a space of at least four dimensions. Problems of visualization arise especially with the geometry of the trajectories for which this 3-torus acts as attracting invariant manifold.

The geometry of the trajectory on the surface of the 2-torus is self-evident; it can be viewed as a line wound into a coil that encloses the torus (see Figure 1). That means, this trajectory is determined by a vector field on the surface of the 2-torus with a slope or pitch which

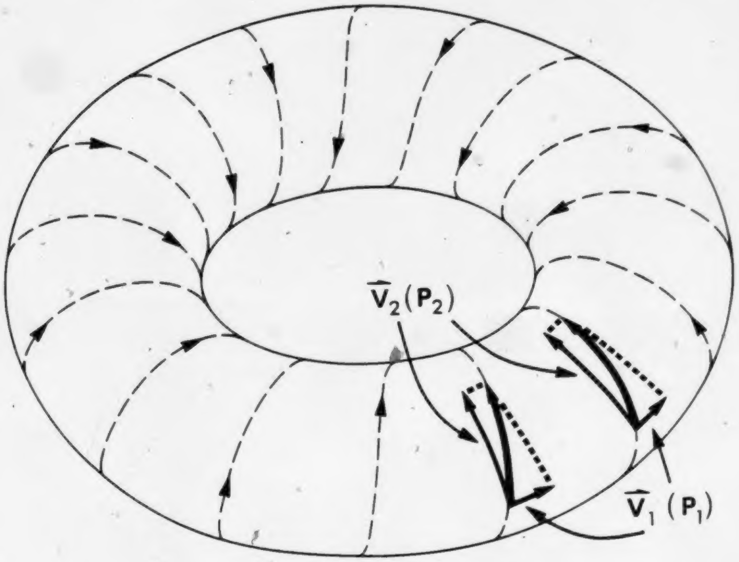

$\vec{V}_2(P_2)$
$\vec{V}_1\ (P_1)$

FIGURE 1. 2-Torus with Trajectory and/or Vector Field on its Surface with Velocity Components $\vec{V}_1$, $\vec{V}_2$ and Periods P_1,P_2

is determined by the ratio of the periods p_1, p_2 arising from the two bifurcations (see Figure 1).

Crossings of further eigenvalue pairs over the imaginary axis cause further Hopf bifurcations and thereby appearance of the appropriate higher-dimensional tori as invariant manifolds.

It is evident that the mathematical description and formulation of the bifurcations following the first one is by far not as straightforward as this very first one, which is a direct application of the Hopf bifurcation theorom on the (linearized) RHS-operator of equations of the type (1.1)*. The second bifurcation does not grow out of a fixed point or a stationary solution, but one that is already periodic in time. Ruelle and Takens therefore introduce a Poincaré map defined on a piece of hypersurface transversal to the first Hopf orbit (see Figure 2). The penetration point of the first Hopf orbit on the Poincaré hypersurface is naturally a fixed point for the Poincaré map. It might of course change position, if the Hopf orbit changes size and shape due to variations in the bifurcation parameter μ. However, if this first orbit destabilizes, so will the fixed point of its associated Poincaré map. If this destabilization is due to a second Hopf bifurcation, this fixed point will degenerate into an invariant set of points sitting on a closed orbit on the Poincaré hypersurface. The eigenvalues of this Poincaré map are in first (linear) approximation the same as the Floquet coefficients of the destabilizing first orbit.

---

*In the case of PDE's even this can become prohibitively difficult, since the analysis of the spectrum of the RHS-operators in order to find functional dependencies between eigenvalues and possible bifurcation parameters is definitely nontrivial.

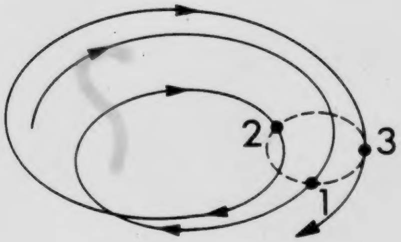


FIGURE 2. Poincaré Map on Plane Transversal to Hopf Orbit

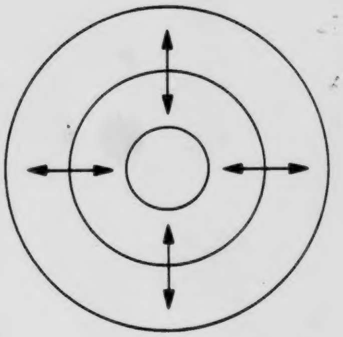


FIGURE 3. Circle Oscillating within Annulus

Most of the technical work in the paper consists of the statements and proofs of the Hopf bifurcation theorems for vector fields (relating to the first bifurcation) and diffeomorphisms (e.g., Poincaré maps used for the second bifurcation) and supporting theorems and lemmas. Bifurcations following the second one which lead to higher-dimensional tori are not being discussed in rigorous mathematical detail in this paper. In the meantime, however, extensive analytic work has been done in this direction by Ioos and Chenciner [7] and Sell [8].

Despite the aforementioned difficulties arising with the complex structure of vector fields and trajectories with higher-dimensional tori as invariant manifolds, there is no reason at this point to assume a breakdown of their deterministic nature and a transition to chaotic and turbulent behaviour. However, in the last section of their paper, Ruelle and Takens show how small perturbations, imposed on vector fields defined on tori of dimension higher than two, can give rise to transitions to vector fields which are not Morse-Smale (see Appendix 1 for definition and explanation) and which have the already mentioned strange attractors. Before this part of the paper is discussed, a few geometric interpretations of projections and Poincare maps of the 3-torus are given.

Advancing the idea of the Poincaré map from the 2-torus (where it arranges the points on a closed orbit, as already described) to a 3-torus one obtains as invariant manifold for the Poincaré map a 2-torus. Similar considerations can be used for projections; the annulus, which arises from the projection of a 2-torus into a plane parallel to the plane, which holds its first circle,* relates to a 2-torus on which the

---

*See footnote on next page.

surface becomes a shell with a certain thickness (as the annulus can be understood as a circle with a certain finite 'line thickness') if a 3-torus is projected into the three dimensional space which hold the cartesian product of its first two circles.* A way to visualize the role of the dimension lost due to projection is to substitute for it by time; in the case of the annulus this leads to a circle oscillating radially within the annulus (see Figure 3) and in the higher-dimensional case of the torus shell one can imagine a 2-torus oscillating within the shell (see Figure 4). This approach works especially well in cases like the present one, where the tori are invariant manifolds of dynamical systems, that is, time dependent systems, and it is evident that the oscillation frequencies would have to be associated with the periods of the circles or closed orbits, respectively, which collapse to line segments under the described projections, i.e. the orbits relating to the second or third bifurcation respectively. Obviously, this idea works also with projections into other planes or three-dimensional spaces than the ones holding the orbits. However, the resulting manifolds would be much more complicated than the annulus and torus shell, since properties of connectedness of the 2- and 3-torus might change drastically under such projections. Especially substitution of time for the missing dimension would lead to rather complicated motions.

Ruelle and Takens define the following vector field z on the 3-torus:

---

*An n-torus embedded into an $m(> n)$-dimensional space can be understood as a noncommuting cartesian product of n circles: $T^n = T_1 \times \ldots \times T_n$. By the terms first circle or first two circles respectively, T_1 or T_1, T_2 respectively is meant here.

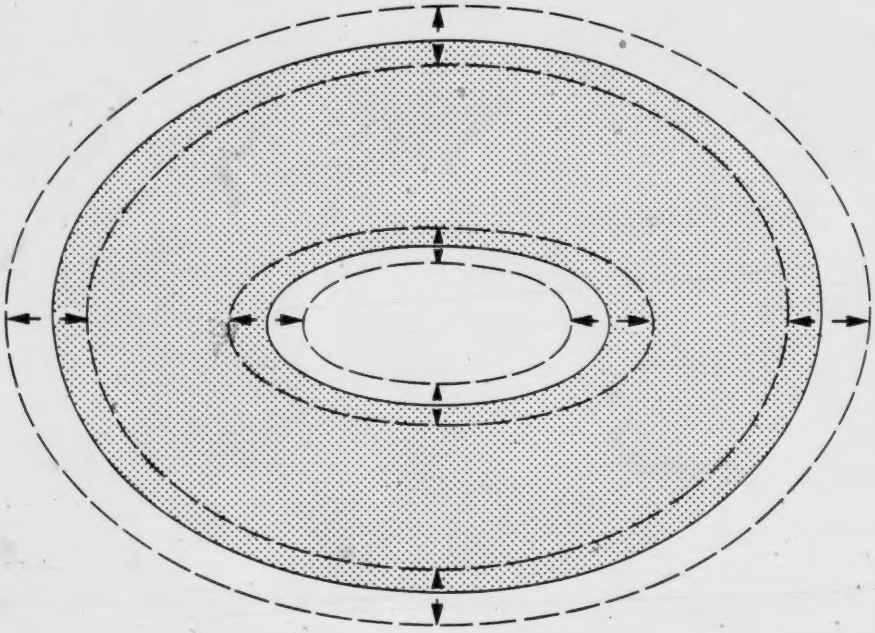


FIGURE 4. 2-Torus Oscillating within Torus Shell

$$z = \frac{\omega_3}{q_1 q_2}\begin{pmatrix} x_1/q_1 \\ x_2/q_2 \\ 1 \end{pmatrix} + \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}$$

$\omega = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}$ is a constant vector field consisting of the three angular velocities associated with the three Hopf-orbits, which generate the 3-torus in their cartesian product.

$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ is the suspension of a horseshoe diffeomorphism on the surface of the 2-torus associated with the ω_1- and ω_2-orbits. That is, x rearranges the points of a disk on the surface of the 2-torus into a horseshoe (as described by S. Smale in [10]; see also Figure 7) within the area originally occupied by the disk. This rearrangement is completed after one return period of ω_3.* In other words, the horseshoe diffeomorphism is here simultaneously a Poincaré map associated with ω_3. Dividing x by an integer, say m, leads to x/m. This means, that the above-mentioned disk will be horseshoe-diffeomorphed once, after a sequence of m return periods (i.e., Poincaré maps) of ω_3 (in difference to the aforementioned case, where x horseshoe-diffeomorphs the disk after one return period). For m large enough, the vector field x/m becomes arbitrarily small. This happens in z, where the role of m is taken over by q_1 and q_2. For q_1 and q_2 large enough, the difference between z and ω can

---

*Clearly, this can be visualized by means of the 2-torus oscillating with period of ω_3 in the shell of the projection of the 3-torus.

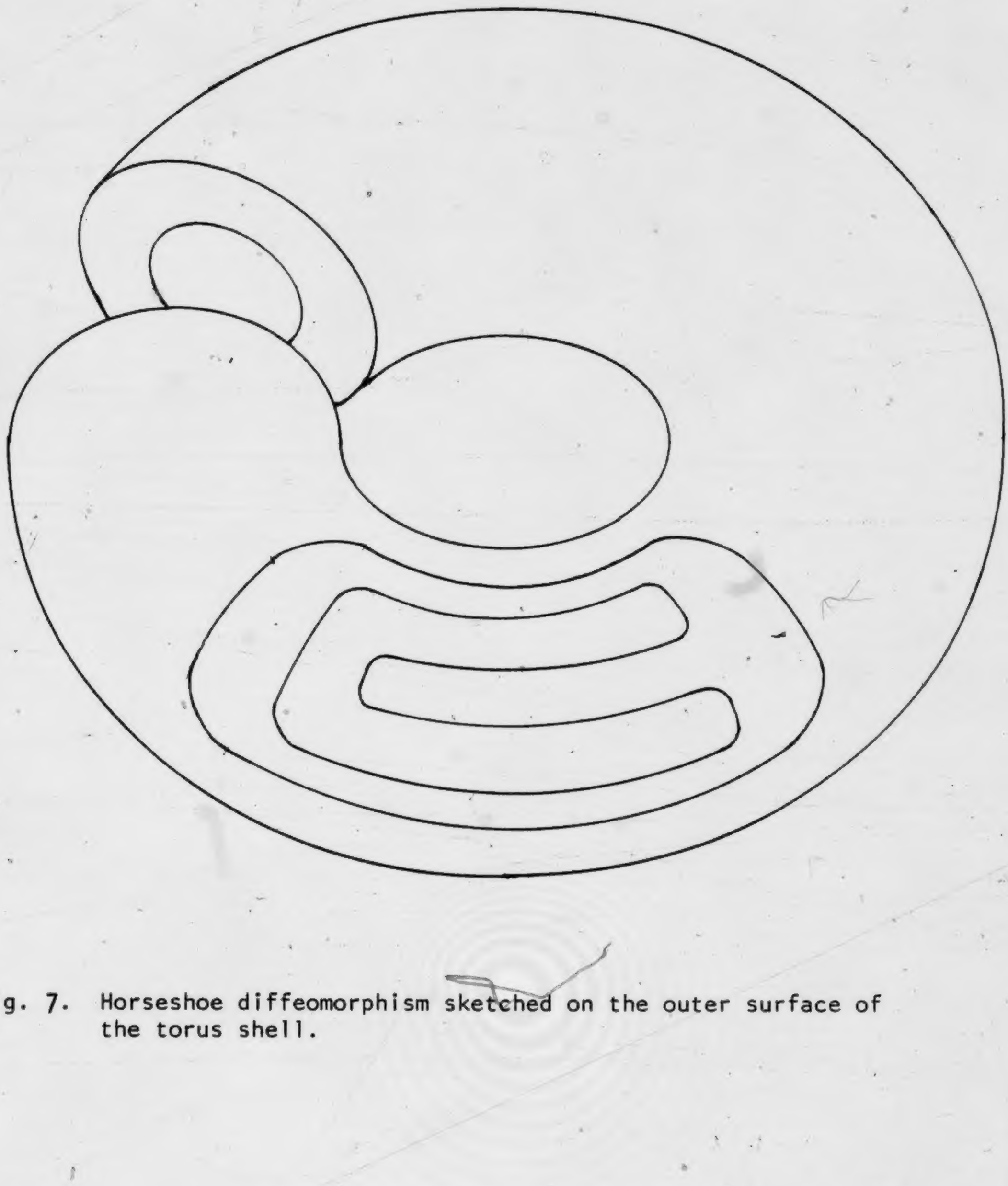


Fig. 7. Horseshoe diffeomorphism sketched on the outer surface of the torus shell.

be made small enough, to put z and ω in any C^2-neighbourhood of each other. The particular structure of z, i.e., the factor ω_3/q_1q_2 and the unit vector in the third component are some of the requirements which guarantee the existence of a Cantor set in the set of non-wandering points of z despite the presence of the part ω. The other requirements are: $0 \leq \omega_1 \leq \omega_2 \leq \omega_3$; $\frac{1}{2} < q_1/q_2 < 2$; $0 < \omega_1/\omega_3 = p_1/q_1 < 1$; $0 < \omega_2/\omega_3 = p_2/q_2 < 1$; p_1, p_2, q_1, q_2 are integers and p_1, q_2, q_1, p_2 have no common divisor. Moreover, the aforementioned disk on T^2 does not occupy any points outside the center square which arises, when the unit square, defining the torus by modulo conditions,* is partitioned into nine equal squares. These modulo conditions also guarantee the formation of a Ω-set which contains a set that is locally the product of a line interval and a Cantor-set. Since the vector field of a Morse-Smale system has a non-wandering set which consists of a finite number of fixed points and periodic orbits only and since in the present system the Ω-set is a subset of the non-wandering set, the present system cannot be Morse-Smale. Besides, the Ω-set does not qualify as a manifold because of the above-explained local product structure and therefore qualifies as a strange attractor.

In the case of a torus with more than three dimensions the horseshoe diffeomorphism is replaced by a diffeomorphism which double-loops the solid 2-torus and maps it into the volume originally occupied by it. This leads to an Ω-set which is locally the product of a Cantor set and a piece of 2-dimensional manifold. Again this is the reason why it fails to be a manifold and is therefore called a strange attractor.

---

*A circle T can be identified with the unit interval I by the mapping: $T \rightarrow \mathbb{R}/\mathbb{Z} = I$ or: $t = r (\mathrm{mod}\ 1)$; $t \in T$; $r \in \mathbb{R}$. Since $T^2 = T \times T$, it can be identified with $I^2 = I \times I$.

All these considerations work equally well if the vector fields under consideration on the higher-dimensional tori are replaced by ones sufficiently close to them (e.g., replace z by z' with $|z - z'|$ sufficiently small in the case of the 3-torus). The described cases are therefore definitely generic (see Appendix A).

3. THE MODEL AND ITS MECHANISM

Informal Description

Looking at the infinite-dimensional state space of the NS-equation defined over a certain domain in fluid space, one realizes that it is spanned by an infinite number of orthogonal 3-dimensional subspaces which relate to the infinite number of points in the fluid space domain with their attached velocity vectors decomposed into their three components. Let us define a vector field over the NS-solution space determined by the accelerations in fluid space according to the NS-equations. The projection of this vector field into one of these 3-dimensional subspaces determines the motion of the velocity vector at the point in fluid space associated with this 3-dimensional subspace. In case the system has undergone two stable Hopf bifurcations its invariant manifold will be the already described 2-torus and its projection into one of these 3-dimensional subspaces will be diffeomorphic to its original in the infinite-dimensional state space (except for nongeneric cases where it might appear as some simply connected 2-dimensional manifold, like the circle might appear as a line element if the direction of projection lies within its plane) and evidently the same goes for the vector field defined on its surface. The tip of the velocity vector associated with this subspace will therefore be moving on a trajectory embedded on the surface of this projected torus. The trajectory itself will be winding around the torus with a pitch that is determined by the ratio of the two frequencies (or periods) stemming from the two bifurcations (see Figure 1).

Advancing this projection idea to the case of a sequence of three bifurcations and the resulting 3-torus in state space, the properties of the projection into a 3-dimensional subspace change drastically. First of all, the projection of the 3-torus from a space with more than three dimensions into a space with three dimensions cannot be diffeomorphic to its original (i.e., the 3-torus). Recalling the discussion of the geometric interpretation of Poincaré maps in Chapter 2, it is evident that for the projection along the axis of the primary orbit into a 3-dimensional subspace orthogonal to it will be the already described torus shell. If the axis of the primary orbit is not orthogonal to the 3-dimensional projection space, the properties of connectedness might change as well. For instance, the hollow space enclosed by the torus shell might vanish locally due to overlapping of walls. (Again this is in analogy to the lower-dimensional case where the opening in the middle of the annulus vanishes due to overlapping effects if the original 2-torus is being projected "sideways" .)

The structure of the vector field within the projection of the

3-torus can easily be derived by using the already discussed method of replacing the dimension lost under projection by time. This leads to a 2-torus oscillating within the torus shell.

The vector field on the surface of this oscillating 2-torus is evidently the one already introduced (see Figure 1) and in addition to it, there is now a third component, perpendicular to the surface of the oscillating 2-torus. This component can be either zero, if the 2-torus is in its maximum or minimum amplitude position, or directed outward (2-torus expanding) or inward (2-torus contracting) (see Figure 8). One realizes therefore two possible vector fields within the torus shell, according to the two possible directions of this third vector component. This situation can degenerate into more than two different vector fields if a projection with locally overlapping walls is considered (see Figure 9). The motion of the velocity vector of a particular point in fluid space after three bifurcations is therefore considerably more complicated than after only two bifurcations.

So far, the motions resulting from the vector fields are still regularly behaved irrespective of whether they relate to two or three bifurcations. Therefore, a mechanism has to be introduced which induces chaotic behaviour.

Looking again at the case of two bifurcations first, one realizes that every one of the projections into the 3-dimensional subspaces is the already discussed 2-torus with a local vector field structure as shown in Figure 10. There, a surface element of this 2-torus with the embedded and surrounding vector field is shown. Suppose now that a small random perturbation, which is short in time

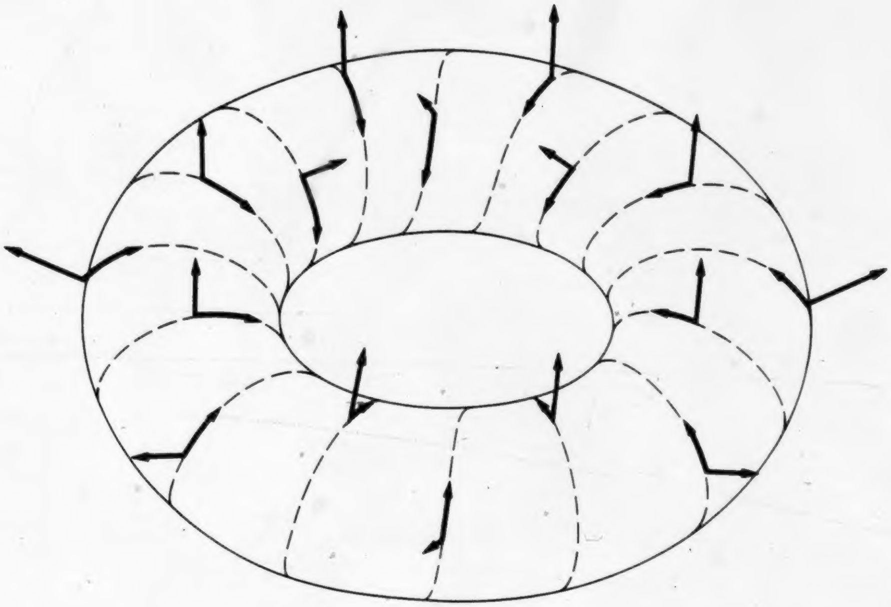


FIGURE 8. Vector Field on Oscillating 2-Torus with Normal Components for Contraction and Expansion

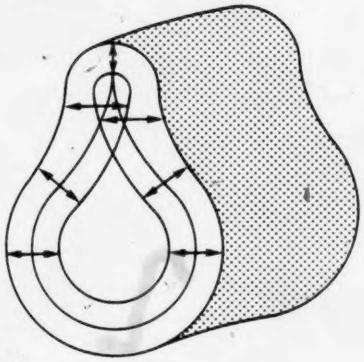


FIGURE 9. Vector Fields on 3-Torus Section with Overlapping Walls Region

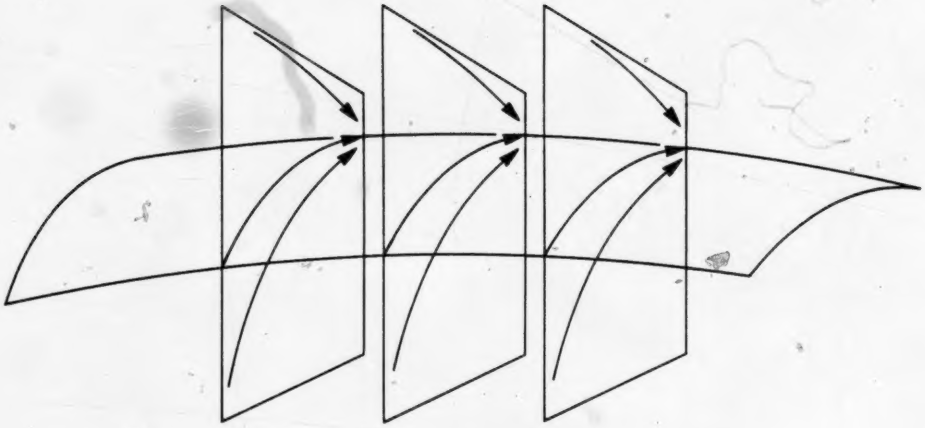


FIGURE 10. Local Vector Field Structure of 2-Torus in 3-Dimensional Subspace

as well, becomes effective at an individual point in fluid space. This will uncouple the subsystem defined by this point from the rest of the system and lead to a displacement of it which has two components: one perpendicular to the toral surface and the other one within it. After the perturbation has relaxed to zero, the subsystem will recouple to the rest system, and the perpendicular component will vanish, since the surrounding vector field will draw the system back into the stable toral surface. The displacement component within the surface, however, will not vanish and the subsystem will follow the vector field on the toral surface again, but slightly displaced sideways w.r.t. its trajectory before the perturbation. Since random perturbations give rise to displacements with directions equally distributed within the toral surface, the mean trajectory of the perturbed subsystem (averaged over a time containing several random perturbations) has to be identical to the trajectory of the unperturbed subsystem. This mechanism holds for all points in fluid space and their state subspaces. The fluid motion after two bifurcations must therefore still be regularly behaved (except for the small random perturbations).

A concise way to explain the effects of the random perturbations on a stable two dimensional manifold would be:

(1) Any random displacement of the system out of the manifold will vanish, since the system will be drawn back into the manifold.

(2) Any sequence of random displacements of the system within the manifold leads to a two-dimensional random-walk motion within the manifold superimposed on the motion due to the vector field on the manifold.

(3) Since the random displacements are very small, the random-walk motion is negligible compared to the motion induced by the vector field.

It is evident, that the above descriptions hold for the original 2-torus in the complete state space as well as for its projections into the individual three-dimensional subspaces.

To investigate the effect of random perturbations on the system after it underwent three bifurcations, consider the motion of the system projected into one of the three-dimensional subspaces. The motion will be determined by one of the possible vector fields in the torus shell, as described earlier. Any sequence of random perturbations acting on the point will result in a three-dimensional random-walk motion superimposed on the vector-field induced motion in the torus shell. This can be concluded by advancing the two-dimensional case previously discussed to three dimensions. For random displacements out of the torus shell again the considerations for the 2-torus with the evident modifications apply.

One way of looking at the dynamics within a three-dimensional subspace is to interpret the velocity components belonging to this subspace as time dependent variables of this subsystem under observation and the velocity components of the remaining system as time-dependent parameters of this subsystem. In other words, one could construct a non-autonomous system of three ODE's for the fluid motion through a particular point in fluid space, if the functional time dependencies of all the other points in fluid space are known. The previously discussed structures of several possible vector fields within the torus shell in the three-dimensional space of the three ODE's would be dependent on the setting of the other velocities in fluid space. That is, which one of the possible vector fields will be realized, is determined by the parameters of the subsystem, i.e., the velocity field in fluid space of the whole system minus the monitored subsystem under observation. One could say therefore, that the structural stability of the vector field momentarily realized in the three-dimensional subspace depends on the momentary setting of its determining parameters, i.e., the velocity field in fluid space of the whole system minus the subsystem under observation. Perturbing these parameters means perturbing the vector field in the subsystem under observation. The main assumption of the model is now that certain combinations of random perturbations on the parameters determining these vector fields can change them (i.e., the parameters) in a way that the vector field determined by them in the subspace changes as well. A sequence of these random perturbation combinations would therefore lead to a sequence of random changes between the realizable vector fields in the subspace, and none of these vector fields would have a lifetime long enough to be considered structurally stable.

Mathematical Formulation

The foregoing considerations can also be expressed by a mathematical formalism. To do so, the NS-equation is being written down in its most general form:

$$\frac{\partial V(t)}{\partial t} = X(V(t)) \qquad (3.1)$$

Introducing a domain D in (physical) fluid space within which Equation (3.1) applies and an infinite-dimensional (mathematical) state space H for Equation (3.1), $V(t)$ can be interpreted physically as a velocity vector field over D at time t or mathematically as a point within H. This point describes the state of velocity of the system within D at time t. $V(t)$ can therefore be defined as the following set:

$$V(t) = \{\vec{v}(r,t);\ r \in D\}, \tag{3.2}$$

with $\vec{v}(r,t)$ as the physical velocity vector at point r and time t.

Accordingly, X can be interpreted physically as the acceleration vector field over D at time t, or mathematically as a vector field over H describing the change of the state of velocity of the system within D at time t, whatever state (or point in H) the system occupies.

X as a vector field over H is time independent; all points in H can be identified with all the possible states of the system, and a change from one particular system state to another does not depend on a particular point in time but on the particular system state only. A particular point V in H will therefore have its time independent "rate-of-change vector" $X(V)$, irrespective of the time the state V realizes.

In its physical interpretation X is time dependent; looking at a particular point in D relates to projecting X from the infinite-dimensional H into the three-dimensional subspace H_i, say, spanned by the velocity components at that particular point $r_i \in D$. This loss of dimensions belonging to the cospace of H_i is automatically compensated for by introduction of time. One could also say that this projection is accompanied with a change of Eulerian coordinates fixed within H to Lagrangean coordinates travelling with the point representing the momentary state of the system. Everything time independent within the Eulerian coordinates has to look time-dependent in the Lagrangean coordinates. X can therefore be written as:

$$X(V(t)) = X(t) = \left\{ \frac{\partial \vec{v}(r,t)}{\partial t} \ ; \ r \in D \right\} \tag{3.3}$$

(physical; Lagrangean coordinates)

or

$$X(V(t)) = \left\{ \frac{\partial V(t)}{\partial t} \ ; \ V(t) \in H \right\}, \tag{3.4}$$

(mathematical; Eulerian coordinates)

whereby the second formulation (3.4) is the more general one relating to Equation (3.1).

Geometrically, the solution space H can be understood as the cartesian product of the three-dimensional orthogonal subspaces H_i ; one for each point $r_i \in D$.

$$H = \underset{r_i \in D}{\times} H_i \ , \quad H_i = \mathbb{R}^{3^*} \ , \quad \vec{v}(r_i,t) \in H_i \tag{3.5}$$

This splitting carries over to the vector field X as a direct sum over all $r_i \in D$:

---

[*] This should be read as: "H_i is a copy of $\mathbb{R}^3$," i.e., transitivity does not apply: $H_i = \mathbb{R}^3$, $H_j = \mathbb{R}^3$, but: $H_i \neq H_j$, for $i \neq j$.

$$X(V(t)) = \bigoplus_{r_i \in D} X_i(V(t)) \tag{3.6}$$

with $X_i(V(t))$ as a three-dimensional vector at point $V(t)$ in H (not in H_i!!) with its components only in the subspace H_i, or, in the physical picture, the acceleration vector at point r_i, "velocity state" V, and time t. Equations (3.2), (3.3) and (3.6) show now that Equation (3.1) can be decomposed accordingly into terms like:

$$\frac{\partial \vec{v}(r_i,t)}{\partial t} = X_i(V(t)) \tag{3.7}$$

In other words, Equation (3.1) can be represented as a system of equations of type (3.7); one for each $r_i \in D$. The further formal "physical decomposition" of Equation (3.7) into three coupled scalar equations for the velocity and acceleration components at r_i is now evident:

$$\frac{\partial v_k}{\partial t} = X_{ik}(V(t)); \quad k = x,y,z . \tag{3.8}$$

Each subsystem of the type (3.7) now depends not only on the velocity $\vec{v}(r_i,t)$ of the point r_i, to which it applies, but (as can be seen by the definition in Equation (3.2)) on all the velocities of the other points $r \in D$ as well, i.e.:

$$\frac{\partial \vec{v}(r_i,t)}{\partial t} = X_i(\vec{v}(r_i,t);\ V(t) \setminus i) \tag{3.9}$$

with:

$$V(t) \setminus i \equiv \left\{ \vec{v}(r,t);\ r \in D - \{r_i\} \right\} \tag{3.10}$$

The right hand side of (3.9) can now be written as:

$$\frac{\partial \vec{v}(r_i,t)}{\partial t} = \vec{X}_{V(t)\setminus i}(\vec{v}(r_i,t)) \tag{3.11}$$

to indicate the role of the elements of $V(t) \setminus i$ as (time dependent) parameters. $\vec{X}_{V(t)\setminus i}$ is a three-dimensional vector field in H_i. Physically, however, it is identical to $X_i(V(t))$, since it performs the same physical function.

To investigate the behaviour of the described system in the presence of a torus as stable invariant manifold, some geometric properties of tori embedded in higher dimensional spaces are being stated. Looking at a circle embedded in such a space in a way that its plane or symmetry axis is not parallel to any of the coordinates spanning the space, it is evident that if two points on this circle have more than one identical coordinate, the two points have to be identical. Moreover, it is evident that amongst the set of possible cartesian coordinate systems for this higher dimensional space, the subset with coordinates parallel to the circle plane or symmetry axis has Lebesgue measure zero. For a 2-torus, the argument holds as well, except that more than two coordinates have to be identical for the rest of the coordinates to be identical as well. This can be stated for higher-dimensional tori as well in the following general way:

Let M be a manifold embedded in $\mathbb{R}^n$ and $P_1^n, P_2^n \in M$.

$$P_1^n = \{x_{i1}\}_{i=1}^n, \quad P_2 = \{x_{i2}\}_{i=1}^n, \tag{3.12}$$

since they are points in $\mathbb{R}^n$ as well.

If there exist subsequences S_1^m, S_2^m within the coordinate strings of P_1^n, P_2^n:

$$S_1^m = \{x_{j1}\}_{j=1}^m , \quad S_2^m = \{x_{j2}\}_{j=1}^m \ni m \leq n; \quad x_{j1} = x_{j2} \forall j \tag{3.13}$$

and if:

$$\left.\begin{array}{l} m > 1 \text{ and } M = T^1 \text{ then } x_{i1} = x_{i2} \forall i \\ m > 2 \text{ and } M = T^2 \text{ then } x_{i1} = x_{i2} \forall i \\ \vdots \\ m > k \text{ and } M = T^k \text{ then } x_{i1} = x_{i2} \forall i \end{array}\right\} \tag{3.14}$$

except for a set of cartesian coordinate systems of Lebesgue measure zero.

Assuming the existence of a circle or a 2-torus as invariant manifold M in H and two points $V(t_1)$, $V(t_2)$ on M:

$$V(t_1),\ V(t_2) \in M \subset H; \quad M = T^1 \text{ or } T^2 \tag{3.15}$$

such that for the restriction or projection Π_i of $V(t_1)$, $V(t_2)$ into one of the three-dimensional subspaces H_i equality holds, i.e.:

$$\underset{\text{(I)}}{\Pi_i(V(t_1))} = \underset{\text{(II)}}{\vec{v}(r_i,t_1)} = \underset{\text{(III)}}{\Pi_i(V(t_2))} = \underset{\text{(IV)}}{\vec{v}(r_i,t_2)} \tag{3.16}$$

and:

$$X(M) = \bigoplus_{r_i \in D} X_i(M) \quad \text{(Whitney Sum)} \tag{3.16a}$$

one can conclude by the foregoing argument, Equation (3.16) (II) = (IV) and by identifying the three components of $\vec{v}(r_i,t_1)$, $\vec{v}(r_i,t_2)$ with the members of the subsequences S_1^m, S_2^m that:

$$V(t_1) = V(t_2) \tag{3.17}$$

or:

$$\vec{X}_{V(t_1)\backslash i}(\vec{v}(r_i,t_1)) = \vec{X}_{V(t_2)\backslash i}(\vec{v}(r_i,t_2)), \quad M = T^1 \text{ or } T^2 \tag{3.18}$$

That means, one and the same point $\vec{v}(r_i)$ fixed within H_i holds one and the same field vector $\vec{X}_{V(t)\backslash i}(\vec{v}(r_i))$ for all times. In other words, if the invariant manifold $M \subset H$ is a circle or 2-torus, the vector field $\vec{X}_{V(t)\backslash i}(\vec{v}(r_i,t))$ associated with it in H_i will be independent of $V(t)\backslash i$, $V(t)$ and thereby independent of time as well. If the invariant manifold $M \subset H$ is a 3-torus one cannot conclude Equation (3.17). Since only three coordinates have to be identical to satisfy Equation (3.16), $m = k = 3$ holds. Therefore, condition (3.14) is not fulfilled, or:

$$\vec{X}_{V(t_1)\backslash i}(\vec{v}(r_i,t_1)) \neq \vec{X}_{V(t_2)\backslash i}(\vec{v}(r_i,t_2)), \quad M = T^3 \tag{3.19}$$

is possible, even if:

$$\vec{v}(r_i,t_1) = \vec{v}(r_i,t_2), \quad t_1 \neq t_2 \tag{3.19a}$$

Therefore, the vector field in H_i associated with an invariant manifold $M = T^3 \subset H$ can change with $V(t)\backslash i = \{\vec{v}(r,t);\ r \in D - \{r_i\}\}$, that is, with all the velocities at all points $r \in D$, except r_i.

One can now introduce the small random perturbation distribution $V_R(t)$ over D at time t with a negligibly small rise time:*

$$V_R(t) = \{\vec{v}_R(r,t);\ r \in D\} \tag{3.20}$$

and impose its reduced form:

$$V_R(t) \setminus i = \left\{\vec{v}_R(r,t);\ r \in D - \{r_i\}\right\} \tag{3.21}$$

upon $V(t) \setminus i$:

$$V(t) \setminus i + V_R(t) \setminus i = \left\{\vec{v}(r,t) + \vec{v}_R(r,t);\ r \in D - \{r_i\}\right\} \tag{3.22}$$

This leads to:

$$\vec{X}_{V(t)\setminus i}(\vec{v}(r_i,t)) \rightarrow \vec{X}_{V(t)\setminus i+V_R(t)\setminus i}(\vec{v}(r_i,t)) \tag{3.23}$$

Assume now for certain perturbation distributions $V_R(t_1)$ the following:

$$V(t_1) \setminus i + V_R(t_1) \setminus i = V(t_2) \setminus i, \tag{3.24}$$

and:

$$\vec{v}(r_i,t_1) = \vec{v}(r_i,t_2), \quad t_1 \neq t_2 \tag{3.24a}$$

Equation (3.18) show now that for $M = T^1$ or T^2 the vector field in H_i remains the same or stable under $V_R(t)$ or $V_R(t) \setminus i$ respectively, but for $M = T^3$ inequality (3.19) shows that the vector field can adopt a structure assigned to a completely different point in time, i.e., t_2, since condition (3.24a) is the same as condition (3.19a), which permits in Equation (3.19) for $M = T^3$. Therefore, the following transition occurs:

---

*The time it takes for the perturbation to reach its maximum.

$$\vec{X}_{V(t_1)\setminus i}(\vec{v}(r_i,t_1)) \rightarrow \vec{X}_{V(t_2)\setminus i}(\vec{v}(r_i,t_2)) = \vec{X}_{V(t_2)\setminus i}(\vec{v}(r_i,t_1) \qquad (3.25)$$

The equation holds since $\vec{v}(r_i,t_2) = \vec{v}(r_i,t_1)$, (more realistic: $\vec{v}(r_i,t_2) \approx \vec{v}(r_i,t_1)$, since the rise time for $V_R(t)$ is finite although negligibly small). This means that, although the individual random perturbation on the individual point in D might be infinitesimally small, certain distributions of them (compatible with Equation (3.24)) imposed on a set of points in D will uncouple a point $r_i \in D$ for the rise time of the perturbation distribution from the rest of the vector field over D by breaking down the vector field $\vec{X}_{V(t_1)\setminus i}(\vec{v}(r_i,t))$ in H_i and then recouple it by realizing a new field $\vec{X}_{V(t_2)\setminus i}(\vec{v}(r_i,t))$ almost randomly selected from a set $\{\vec{X}_{V(t)\setminus i}\}$, say, permissible by Equation (3.11). This new vector field $\vec{X}_{V(t_2)\setminus i}(\vec{v}(r_i,t))$ might not necessarily be C^0 equivalent to $\vec{X}_{V(t_1)\setminus i}(\vec{v}(r_i,t))$, which would mean that some of the elements $\vec{X}_{V(t)\setminus i}$ of the set $\{\vec{X}_{V(t)\setminus i}\}$ are not structurally stable. Clearly, a sequence of such random distributions leads to chaotic behaviour at the individual point $r_i \in D$ and this therefore generalized directly to the entire D.

To complete this description, some consideration is given to the already mentioned set of cartesian coordinate systems of Lebesgue measure zero, for which the relations (3.14) do not hold. That is, if the invariant manifold is a k-torus, two different points on it can be in more than k coordinates identical. The random perturbed behaviour of the system on a 3-torus will essentially be the same as already described, except for a possible change in the set of selectable vector fields $\{\vec{X}_{V(t)\setminus i}\}$ on H_i. For a circle or 2-torus, however, for some set L, say, of subspaces H_i (with Lebesgue measure zero), inequality (3.19) instead of Equation (3.18) will hold. That means, for one or two bifurcations some points r_i (associated with the $H_i \in L$) can show chaotic behaviour. At first glance, it appears as if this kind of behaviour can be induced deliberately by the way the coordinate system spanning H is being set up or rotated. One might be able to select one belonging to the set (of Lebesgue measure zero) for which the relations (3.14) do not hold at will. However, H is spanned by the H_i (see Equation (3.5)) and the structure of any combination of them is determined by the structure of the $r_i \in D$, i.e., the geometry of D. The three-dimensional cartesian coordinate systems spanning the individual H_i's can be rotated in any way without changing the applicability or inapplicability of the relations (3.14) for the individual r_i's. To obtain such an effect, the individual H_i's would have to be spanned by velocity components belonging to different points $r \in D$ and could therefore not be associated with one particular r_i, but two or three. Obviously, this would be meaningless within this model.

PART 2

Technical Realization and Approximation of the Model

As already mentioned in the introduction, there are two levels of analysis of the fluid dynamic equations. One level, the qualitative analysis and description of the mechanism for turbulence was the subject of the previous chapter. In the present part, the approach is more application-oriented in that a technical basis for computer simulation is developed. Before the main problem of discretization of the fluid domain D is addressed, a particular type of NS-equation is derived, which permits one to keep a certain generality w.r.t. the selection of various flow structures.

4. THE NS-EQUATIONS FOR PERTURBATIONS

For a velocity vector $\vec{U}$ at point r and time t, the NS-equation is written as:

$$\frac{\partial\vec{U}(r,t)}{\partial t} = -\vec{U}(r,t)[\vec{\nabla} : \vec{U}(r,t)] + \nu\Delta\vec{U}(r,t) - \vec{\nabla}p(r,t) \qquad (4.1)$$

with

$$\vec{a}[\vec{\nabla} : \vec{b}] \equiv \begin{pmatrix} a_x \frac{\partial b_x}{\partial x} + a_y \frac{\partial b_x}{\partial y} + a_z \frac{\partial b_x}{\partial z} \\ a_x \frac{\partial b_y}{\partial x} + a_y \frac{\partial b_y}{\partial y} + a_z \frac{\partial b_y}{\partial z} \\ a_x \frac{\partial b_z}{\partial x} + a_y \frac{\partial b_z}{\partial y} + a_z \frac{\partial b_z}{\partial z} \end{pmatrix} \qquad (4.2)$$

as the advective term, and

ν = kinematic viscosity,

p(r,t) = pressure at point r and time t.

For convenience, an advective operator A can be introduced:

$$A[\vec{a}\vec{b}] \equiv \vec{a}[\vec{\nabla} : \vec{b}] + \vec{b}[\vec{\nabla} : \vec{a}] \tag{4.3}$$

Clearly, this operator is symmetric and bilinear in $\vec{a}$ and $\vec{b}$. Equation (4.1) becomes now

$$\frac{\partial\vec{U}(r,t)}{\partial t} = -\frac{A}{2}[\vec{U}(r,t)\vec{U}(r,t)] + \nu\Delta\vec{U}(r,t) - \vec{\nabla}p(r,t) \tag{4.4}$$

and holds also, if $\vec{U}$ develops a perturbation $\vec{V}_1$:

$$\vec{U}(r,t) \rightarrow \vec{U}(r,t) + \vec{V}_1(r,t) \tag{4.5}$$

$$\frac{\partial(\vec{U}+\vec{V}_1)}{\partial t} = -\frac{A}{2}[(\vec{U}+\vec{V}_1)(\vec{U}+\vec{V}_1)] + \nu\Delta(\vec{U}+\vec{V}_1) - \nabla\vec{p} \tag{4.6}$$

(For reasons of conciseness, the dependencies (r,t) are not written out in Equation (4.6)).

Since A is bilinear, Equation (4.4) can be subtracted from Equation (4.6) to obtain an equation for the perturbation $\vec{V}_1$:

$$\frac{\partial\vec{V}_1}{\partial t} = -A[\vec{U}\,\vec{V}_1] - \frac{A}{2}[\vec{V}_1\vec{V}_1] + \nu\Delta\vec{V}_1 \tag{4.7}$$

Using the same method for a perturbation $\vec{V}_2$ on top of $\vec{V}_1$:

$$\vec{V}_1 \rightarrow \vec{V}_1 + \vec{V}_2 \tag{4.8}$$

leads to an equation for $\vec{V}_2$:

$$\frac{\partial\vec{V}_2}{\partial t} = -A[\vec{U}\vec{V}_2] - A[\vec{V}_1\vec{V}_2] - \frac{A}{2}[\vec{V}_2\vec{V}_2] + \nu\Delta\vec{V}_2 . \tag{4.9}$$

A third perturbation $\vec{V}_3$ on top of $\vec{V}_2$:

$$\vec{V}_2 \to \vec{V}_2 + \vec{V}_3 \tag{4.10}$$

leads to an equation for $\vec{V}_3$:

$$\frac{\partial \vec{V}_3}{\partial t} = -A[\vec{U}\vec{V}_3] - A[\vec{V}_1\vec{V}_3] - A[\vec{V}_2\vec{V}_3] - \frac{A}{2}[\vec{V}_3\vec{V}_3] + \nu\Delta\vec{V}_3 \tag{4.11}$$

Naturally, these equations for the perturbations can also be obtained without the introduction of the advective operator A. Their derivation, however would be rather lengthy.

A more accurate and realistic approach would require the introduction of a pressure perturbation as well, e.g.:

$$p \to p + p_1 \tag{4.12}$$

However, this would require an evolution equation for the pressure. Unfortunately, such an equation does not exist. Moreover, it is evident that this neglect is legitimate for the assumption of isotropic pressure perturbations, since their gradient vanishes.

Equations (4.7), (4.9), (4.11) determine the stability of a given solution by solving for the time development of a perturbation. Assume a given flow field $\vec{U}(r,t)$ as a solution of Equation (4.4) and substitute it into Equation (4.7). The perturbation $\vec{V}_1(r,t)$ will vanish for stable $\vec{U}(r,t)$ and grow unbounded or become periodic for $\vec{U}(r,t)$ unstable or undergoing a Hopf bifurcation. $\vec{U}(r,t) + \vec{V}_1(r,t)$ can now be substituted into Equation (4.9) to obtain the perturbation $\vec{V}_2(r,t)$ and $\vec{U}(r,t) + \vec{V}_1(r,t) + \vec{V}_2(r,t)$ in Equation (4.11) works accordingly with $\vec{V}_3(r,t)$.

This particular approach was motivated by the availability of a FORTRAN program which investigates (Hopf) bifurcational behaviour of ODE's. The program can be used on a discretized version of Equation (4.7), i.e., a system of ODE's, to test a given stationary (discretized) flow field $\vec{U}(r)$ for Hopf bifurcations. If a Hopf bifurcation occurs, $\vec{U}(r)$ and the periodic solution $\vec{V}_1(r,t)$ can be tested in Equation (4.9) (discretized) for a second Hopf bifurcation which leads to a 2-torus as invariant manifold in solution space. After that, Equation (4.11) can be used to test for the third Hopf bifurcation. Evidently, this method "lives" on the particular structure of the advective terms in the NS-equation or Equations (4.7), (4.9), (4.11) respectively. It is this term which permits the construction of evolution equations for perturbations without eliminating the underlying flow out of the equation.

However, the nonautonomous nature of Equations (4.9) and (4.11) causes problems for the application of the bifurcation program, since time-dependent functions instead of stationary distributions have to be tested for bifurcations.* An attempt to deal with this problem is described later in the chapter.

Discretization of the NS-equations**

To obtain numerical approximate solutions for equations of the type (4.7), (4.9), (4.11), they have to be approximated by a system of ODE's. Such a system is obtained by selecting a discrete finite set S, say, of points r_s from the fluid domain D and constructing a subsystem of three ODE's for the three components of the unknown perturbation velocities $\vec{V}(r_s)$ at each $r_s \in S$. These different subsystems have to be coupled mutually to account for the interaction between the different $r_s \in S$. In the next section such a system of ODE's for Equation (4.7) is derived using finite element techniques and the Galerkin method. At the same time, some of the theory and motivation behind this approach is explained. For instance, it is being shown how incompressibility and Dirichlet boundary conditions can be obtained by appropriate construction of basis functions. Furthermore an expansion from the ODE-system for Equation (4.7) to systems for Equations (4.9) and (4.11) is performed.

---

*See also Chapter 2

**See Appendix D for an introduction to previous works.

5. THE GALERKIN METHOD

One of the most widely used and successful discretization-approximation techniques is the Galerkin method. It is applied here in connection with some kind of function space representation, i.e., the functions $\vec{U}$, $\vec{V}_1$ appearing in the Equation (4.7) are being represented as vectors in the infinite-dimensional function space H introduced in Chapter 3:

$$\frac{\partial}{\partial t}\,|\vec{V}_1> = -|\vec{U}[\vec{\nabla} : \vec{V}_1]> - \,|\vec{V}_1[\vec{\nabla} : \vec{U}]> + \nu|\Delta\vec{V}_1> - \,|\vec{V}_1[\vec{\nabla} : \vec{V}]>^{*} \qquad (5.1)$$

---

[*] For reasons of conciseness the spatial and temporal dependencies (r,t) are not written out.

This is Equation (4.7) written in function-space notation without the use of the advective operator A. The symbol $|>$ is borrowed from the formalism used in quantum theory and denotes a point or vector in function space, i.e., in H here.

The "subdecomposition" into fluid space components is also evident, e.g.:

$$|\vec{U}> \equiv \begin{pmatrix} |U_1> \\ |U_2> \\ |U_3> \end{pmatrix} \tag{5.2}$$

Again following the quantum mechanical formalism, the identity operator I is introduced:

$$I \equiv \sum_{r \in D} |\Delta_r><\Delta_r| \, . \tag{5.3}$$

Δ_r is the three-dimensional Dirac delta function of the point $r \in D$ (see Appendix B). The evident interpretation of definition (5.3) within the concept of function spaces or infinite-dimensional vector spaces is that of a point or vector $|v>$, say, being decomposed by I into the components parallel to the coordinates of the infinite-dimensional system defined by the orthogonal vectors $|\Delta_r>$, i.e.,

$$I|V> = \sum_{r \in D} |\Delta_r><\Delta_r|V> \tag{5.4}$$

The $<\Delta_r|V>$ can be understood as the coordinates or component values of $|V>$ within the system spanned by the $|\Delta_r>$, $r \in D$, i.e., the scalar products between the $|\Delta_r\rangle$ and $|V\rangle$.*

In order to separate the perturbation velocities $\vec{V}_1$ out to the right the identity operator I will be applied to the left of each of these perturbations in Equation (5.1). After that, a finite set S of points r_n is selected from the entire domain D. This restricts the calculations of the perturbation field $\vec{V}_1(r,t)$, $r \in D$ to the calculations of the discretely distributed $\vec{V}_1(r_n,t)$, $r_n \in S$; and reduces the partial differential NS-equations (4.7) to a system $3 \times N$ ODE's, where N is the cardinality of S. (This includes the fluid space subdecomposition according to Equation (5.2) as well.) The spatial derivatives introduced therein by the Del and Laplace operators could now be replaced by finite-difference approximations constructed of elements like e.g.:

$$\frac{\partial f(r_n)}{\partial r} \approx \frac{f(r_{n+1})-f(r_{n-1})}{2h} \tag{5.5}$$

with:

r_n = selected point
$r_{n\pm1}$ = selected points in neighbourhood of r_n
h = spatial distance between r_n, r_{n+1} or r_n, r_{n-1} resp.**

This and various other finite-difference approximations will not preserve the incompressibility condition $\operatorname{div} \vec{U} = 0$ of an initial velocity field $\vec{U}$ in the subsequent numerical iterations, regardless of whether they develop the evolution NS-equation in time or just iterate a steady state NS-equation via algorithms towards a higher accuracy. One therefore depends on a method that preserves incompressibility. Such a method is most efficiently dealt with and explained in the function or state space context.

---

*Introduction of the duality concept between bra ($\langle|$) and ket ($|\rangle$) vectors is not necessary, since only real-valued functions are used in this context.

**Naturally a discretization with equispacing is assumed.

The state space of the NS-equations is a subspace of the function space represented by the Dirac functions as basis functions Δ_r, $r \in D$. That is, application of the identity operator I defined in Equation (5.3) will represent a NS-solution as a point in a space spanned by the vectors $|\Delta_r\rangle$, $r \in D$. Within this state space there is a subspace V in which the condition $\operatorname{div} \vec{U} = 0$, $\forall |\vec{U}\rangle \in V$ holds. The obvious choice for a set of basis functions spanning such a space is a set of scalar functions $W_x(y)$, $x, y \in D$ which, when multiplied with a velocity field $\vec{U}$ over D yield a divergence-free vector field, i.e.:

$$\operatorname{div}(W_x(y) \cdot \vec{U}(x)) = 0, \tag{5.6}$$

where the derivatives are taken w.r.t. y and the velocity $\vec{U}(x)$ is the value of the field taken at the point x; that is, x is not to be understood as a variable in Equation (5.6). (It is evident that these basis functions don't have to be orthogonal to each other such as the Dirac delta functions Δ_r. Their supports within D may overlap.)

Replacing each Dirac-type basis function Δ_x, $x \in D$ by $W_x(y)$, $x, y \in D$ leads to the following representation of the solution $\vec{U}$:

$$\begin{aligned}\sum_{x\in D} |W_x(y)\rangle\langle W_x(y)|\vec{U}\rangle &= \sum_{x\in D} W_x(y) \int_D W_x(y)\vec{U}(y)dy \\ &= \sum_{x\in D} W_x(y)\,\overline{\vec{U}}\,, \end{aligned} \qquad (5.7)$$

i.e., a linear combination of an infinite number of basis functions $W_x(y)$ with some type of mean values $\overline{\vec{U}}$ of $\vec{U}$ as coefficients.

If the solution $\vec{U}$ had nonzero divergence, application of Equation (5.7) would project $\vec{U}$ into a divergence-free or solenoidal subspace of the solution space. Without knowledge of the entire velocity field $\vec{U}$, the means $\overline{\vec{U}}$ cannot be calculated and therefore, only the point values $\vec{U}(r_n)$ at the points $r_n \in S$ selected from D are being used. This discretization produces the following approximation to Equation (5.7):

$$\begin{aligned}\sum_{r_n\in S} |W_{r_n}(y)\rangle\langle \Delta_{r_n}|\vec{U}\rangle &= \sum_{r_n\in S} W_{r_n}(y) \int_D \Delta_{r_n}\vec{U}(y)dy \\ &= \sum_{r_n\in S} W_{r_n}(y)\vec{U}(r_n) \end{aligned} \qquad (5.8)$$

Application of the herein defined "discretization-approximation-projection" operator

$$I_{dap} \equiv \sum_{r_n\in S} |W_{r_n}(y)\rangle\langle \Delta_{r_n}| \qquad (5.9)$$

to the perturbation velocities $\vec{V}_1$ in Equation (5.1) leads to a system where the spatial derivatives are taken analytically on the W_{r_n}'s instead of using some finite difference approximation over discrete points as shown in Equation (5.5)

Applying therefore I_{dap} on $|\vec{V}_1\rangle$ in Equation (5.1) from the left gives:

$$\begin{aligned}\frac{\partial}{\partial t}\sum_{r_n\in S}|W_{r_n}(y)\rangle\langle\Delta_{r_n}|\vec{V}_1\rangle = &-\sum_{r_n\in S}|\vec{U}[\vec{\nabla}:W_{r_n}(y)]\rangle\langle\Delta_{r_n}|\vec{V}_1\rangle\\ &-\sum_{r_n\in S}|[\vec{\nabla}:\vec{U}]W_{r_n}(y)\rangle\langle\Delta_{r_n}|\vec{V}_1\rangle\\ &+\nu\sum_{r_n\in S}|\Delta\cdot W_{r_n}(y)\rangle\langle\Delta_{r_n}|\vec{V}_1\rangle\\ &-\sum_{r_n,r_m\in S}|W_{r_n}(y)\vec{\nabla}W_{r_m}(y)\rangle\langle\Delta_{r_n}|\vec{V}_1\rangle\langle\Delta_{r_m}|\vec{V}_1\rangle \qquad (5.10)\end{aligned}$$

here:

$$\vec{U}[\vec{\nabla}:W_{r_n}(y)] \equiv \begin{pmatrix} U_1\dfrac{\partial W_{r_n}}{\partial y_1}+U_2\dfrac{\partial W_{r_n}}{\partial y_1}+U_3\dfrac{\partial W_{r_n}}{\partial y_1}\\[2ex] U_1\dfrac{\partial W_{r_n}}{\partial y_2}+U_2\dfrac{\partial W_{r_n}}{\partial y_2}+U_3\dfrac{\partial W_{r_n}}{\partial y_2}\\[2ex] U_1\dfrac{\partial W_{r_n}}{\partial y_3}+U_2\dfrac{\partial W_{r_n}}{\partial y_3}+U_3\dfrac{\partial W_{r_n}}{\partial y_3}\end{pmatrix} \qquad (5.10a)$$

$$[\vec{\nabla}:\vec{U}]W_{r_n}(y) \equiv W_{r_n}(y)\begin{pmatrix}\dfrac{\partial U_1}{\partial y_1}+\dfrac{\partial U_1}{\partial y_2}+\dfrac{\partial U_1}{\partial y_3}\\[2ex] \dfrac{\partial U_2}{\partial y_1}+\dfrac{\partial U_2}{\partial y_2}+\dfrac{\partial U_2}{\partial y_3}\\[2ex] \dfrac{\partial U_3}{\partial y_1}+\dfrac{\partial U_3}{\partial y_2}+\dfrac{\partial U_3}{\partial y_3}\end{pmatrix} \qquad (5.10b)$$

Clearly, this follows directly from Equation (4.2). The nonlinear term is derived as follows:

$$|\vec{V}_1[\vec{\nabla} : \vec{V}_1]> \xrightarrow{I_{dap}*} \sum_{r_n \epsilon S} |W_{r_n}(y)><\Delta_{r_n}|\vec{V}_1> \sum_{r_m \epsilon S} |\vec{\nabla} W_{r_m}(y)><\Delta_{r_m}|\vec{V}_1>$$

$$= \sum_{r_n, r_m \epsilon S} |W_{r_n}(y)><\Delta_{r_n}|\vec{V}_1> | \nabla W_{r_m}(y)><\Delta_{r_m}|\vec{\ }_1>$$

$$= \sum_{r_n, r_m \epsilon S} |W_{r_n}(y)\vec{\nabla} W_{r_m}(y)><\Delta_{r_n}|\vec{V}_1><\Delta_{r_m}|\vec{V}_1>$$

Next, Equation (5.10) is multiplied from the left with the operators:

$$P_0 \equiv <W_{r_o}(y) | \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = <W_{r_o}(y)|I_3; \quad r_o \epsilon S \qquad (5.11)$$

to obtain a system of $3 \times N\cdot$ ODE's:

$$\frac{\partial}{\partial t} \sum_{r_n \epsilon S} <W_{r_o}(y) I_3 | W_{r_n}(y)><\Delta_{r_n}|\vec{V}_1>$$

$$= - \sum_{r_n \epsilon S} <W_{r_o}(y) I_3 | \vec{U}[\vec{\nabla} : W_{r_n}(y)]><\Delta_{r_n}|\vec{V}_1>$$

$$- \sum_{r_n \epsilon S} <W_{r_o}(y) I_3 | [\vec{\nabla} : \vec{U}] W_{r_n}(y)><\Delta_{r_n}|\vec{V}_1>$$

$$+ \nu \sum_{r_n \epsilon S} <W_{r_o}(y) I_3 | \Delta W_{r_n}(y)><\Delta_{r_n}|\vec{V}_1>$$

$$- \sum_{r_n, r_m \epsilon S} <W_{r_o}(y) I_3 | W_{r_n}(y) \vec{\nabla} W_{r_m}(y)><\Delta_{r_n}|\vec{V}_1><\Delta_{r_m}|\vec{V}_1> ; \; r_o \epsilon S. \qquad (5.12)$$

This is a system of known coefficient matrices and vectors consisting of the unknown perturbation velocities $\vec{V}_1$ at the points $r_o \epsilon$ S. The 3×3 unit matrix I_3 controls proper subdecomposition

---

[*] The arrow translates as: "application of I_{dap} yields".

of the coefficients into their fluid space components.

To show the structure of these matrix elements or coefficients, especially in their fluid-space subdecomposition, they are being written out explicitly as integrals and at the same time technical abbreviations are introduced below.

$$W_{on} \equiv \langle W_{r_o}(y) I_3 | W_{r_n}(y) \rangle = \int_D W_{r_o}(y) W_{r_n}(y)\,dy \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.13}$$

$$W_{on}^{U\nabla} \equiv \langle W_{r_o}(y) I_3 | \vec{U}[\vec{\nabla} : W_{r_n}(y)] \rangle$$

$$= \int_D W_{r_o}(y) \left[U_1(y) \frac{\partial W_{r_n}(y)}{\partial y_1} + U_2(y) \frac{\partial W_{r_n}(y)}{\partial y_2} + U_3(y) \frac{\partial W_{r_n}(y)}{\partial y_3} \right] dy \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.14}$$

$$W_{on}^{\nabla:U} \equiv \langle W_{r_o}(y) I_3 | [\vec{\nabla} : \vec{U}] W_{r_n}(y) \rangle$$

$$= \begin{bmatrix} \int_D W_{r_o}(y) W_{r_n}(y) \left[\frac{\partial U_1}{\partial y_1}\right] dy & \int_D W_{r_o}(y) W_{r_n}(y) \left[\frac{\partial U_1}{\partial y_2}\right] dy & \int_D W_{r_o}(y) W_{r_n}(y) \left[\frac{\partial U_1}{\partial y_3}\right] dy \\ \int_D W_{r_o}(y) W_{r_n}(y) \left[\frac{\partial U_2}{\partial y_1}\right] dy & \int_D W_{r_o}(y) W_{r_n}(y) \left[\frac{\partial U_2}{\partial y_2}\right] dy & \int_D W_{r_o}(y) W_{r_n}(y) \left[\frac{\partial U_2}{\partial y_3}\right] dy \\ \int_D W_{r_o}(y) W_{r_n}(y) \left[\frac{\partial U_3}{\partial y_1}\right] dy & \int_D W_{r_o}(y) W_{r_n}(y) \left[\frac{\partial U_3}{\partial y_2}\right] dy & \int_D W_{r_o}(y) W_{r_n}(y) \left[\frac{\partial U_3}{\partial y_3}\right] dy \end{bmatrix}$$

$$= \int_D W_{r_o}(y) W_{r_n}(y) \begin{bmatrix} \frac{\partial U_1}{\partial y_1} & \frac{\partial U_1}{\partial y_2} & \frac{\partial U_1}{\partial y_3} \\ \frac{\partial U_2}{\partial y_1} & \frac{\partial U_2}{\partial y_2} & \frac{\partial U_2}{\partial y_3} \\ \frac{\partial U_3}{\partial y_1} & \frac{\partial U_3}{\partial y_2} & \frac{\partial U_3}{\partial y_3} \end{bmatrix} dy \tag{5.15}$$

$$W^{\Delta}_{on} \equiv \langle W_{r_o}(y) I_3 | \Delta W_{r_n}(y) \rangle$$

$$= \int_D W_{r_o}(y) \left[\frac{\partial^2 W_{r_n}(y)}{\partial y_1^2} + \frac{\partial^2 W_{r_n}(y)}{\partial y_2^2} + \frac{\partial^2 W_{r_n}(y)}{\partial y_3^2} \right] dy \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (5.16)$$

For the matrix elements relating to the nonlinear term

$$W^{\nabla}_{onm} \equiv \langle W_{r_o}(y) I_3 | W_{r_n}(y) \vec{\nabla} W_{r_m}(y) \rangle \qquad (5.17)$$

a more detailed analysis is appropriate. On function space level the term is written as:

$$\sum_{r_n, r_m \in S} \langle W_{r_o}(y) I_3 | W_{r_n}(y) \overline{\nabla} W_{r_m}(y) \rangle \langle \Delta_{r_n} | \vec{V}_1 \rangle \langle \Delta_{r_m} | \vec{V}_1 \rangle$$

The terms $\langle W_{r_o}(y) I_3 | W_{r_n}(y) \vec{\nabla} W_{r_m}(y) \rangle$ are elements of a $N \times N^2$ matrix (o counts rows and $n \cdot m$ counts columns), whereas the $\langle \Delta_{r_n} | \vec{V}_1 \rangle \langle \Delta_{r_m} | \vec{V}_1 \rangle$ are components of a N^2-dimensional column vector. Since $\langle \Delta_{r_n} | \vec{V}_1 \rangle \langle \Delta_{r_m} | \vec{V}_1 \rangle = \langle \Delta_{r_m} | \vec{V}_1 \rangle \langle \Delta_{r_n} | \vec{V}_1 \rangle$ these N^2 columns and vector components can be reduced to $N(N+1)/2$ columns or vector components respectively.

On fluid-space level one has:

$$\underbrace{\left(W_{r_o}(y) W_{r_n}(y) \frac{\partial W_{r_m}(y)}{\partial y_1}, W_{r_o}(y) W_{r_n}(y) \frac{\partial W_{r_m}(y)}{\partial y_2}, W_{r_o}(y) W_{r_n}(y) \frac{\partial W_{r_m}(y)}{\partial y_3} \right)}_{\substack{\text{3 dimensional row vector} \\ \text{(known coefficients)}}} \underbrace{\begin{bmatrix} V_{11}(r_n)V_{11}(r_m) & V_{11}(r_n)V_{12}(r_m) & V_{11}(r_n)V_{13}(r_m) \\ V_{12}(r_n)V_{11}(r_m) & V_{12}(r_n)V_{12}(r_m) & V_{12}(r_n)V_{13}(r_m) \\ V_{13}(r_n)V_{11}(r_m) & V_{13}(r_n)V_{12}(r_m) & V_{13}(r_n)V_{13}(r_m) \end{bmatrix}^{*}}_{\substack{3 \times 3 \text{ product matrix} \\ \text{(unknown velocities)}}}$$

---

[*] The following notation is used:

e.g., $V_{13}(r_n)$ means: the third vector component of the perturbation velocity $\vec{V}_1$ at point (r_n).

Recall also $\langle \Delta_{r_n} | \vec{V}_1 \rangle = \vec{V}_1(r_n)$.

Since each coefficient matrix element of the linear terms decomposes into a 3×3 matrix on fluid-space level, one actually has a $3N \times 3N$ coefficient matrix for each linear term. To multiply the inverse of the $3N \times 3N$ coefficient matrix of the left-hand side of Equation (5.12) with the right hand side, one needs for the nonlinear terms just as for the linear terms a coefficient matrix with $3N$ rows. Therefore the three-dimensional coefficient row vector of the above expression has to be transformed into a matrix with three rows, and the 3×3 matrix of the products of the perturbation velocity components into a column vector of the same dimension as columns in the matrix-transformed 3-dimensional row coefficient vector. The definition of a dyadic product between the 3×3 unit matrix I_3 and the 3-dimensional row coefficient vector casts the situation of fluid space level into a 3×9 coefficient matrix and a 9-dimensional column vector consisting of the 3 columns of the 3×3 product matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} : \left(W_{r_o}(y) W_{r_n}(y) \frac{\partial W_{r_m}(y)}{\partial y_1}, W_{r_o}(y) W_{r_n}(y) \frac{\partial W_{r_m}(y)}{\partial y_2}, W_{r_o}(y) W_{r_n}(y) \frac{\partial W_{r_m}(y)}{\partial y_3} \right)$$

$$\times \begin{bmatrix} V_{11}(r_n)V_{11}(r_m) & V_{11}(r_n)V_{12}(r_m) & V_{11}(r_n)V_{13}(r_m) \\ V_{12}(r_n)V_{11}(r_m) & V_{12}(r_n)V_{12}(r_m) & V_{12}(r_n)V_{13}(r_m) \\ V_{13}(r_n)V_{11}(r_m) & V_{13}(r_n)V_{12}(r_m) & V_{13}(r_n)V_{13}(r_m) \end{bmatrix} =$$

$$= W_{r_o}(y)W_{r_n}(y)\begin{bmatrix} \frac{\partial W_{r_m}(y)}{\partial y_1} & \frac{\partial W_{r_m}(y)}{\partial y_2} & \frac{\partial W_{r_m}(y)}{\partial y_3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial W_{r_m}(y)}{\partial y_1} & \frac{\partial W_{r_m}(y)}{\partial y_2} & \frac{\partial W_{r_m}(y)}{\partial y_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial W_{r_m}(y)}{\partial y_1} & \frac{\partial W_{r_m}(y)}{\partial y_2} & \frac{\partial W_{r_m}(y)}{\partial y_3} \end{bmatrix} \begin{bmatrix} V_{11}(r_n)\ V_{11}(r_m) \\ V_{12}(r_n)\ V_{11}(r_m) \\ V_{13}(r_n)\ V_{11}(r_m) \\ V_{11}(r_n)\ V_{12}(r_m) \\ V_{12}(r_n)\ V_{12}(r_m) \\ V_{13}(r_n)\ V_{12}(r_m) \\ V_{11}(r_n)\ V_{13}(r_m) \\ V_{12}(r_n)\ V_{13}(r_m) \\ V_{13}(r_n)\ V_{13}(r_m) \end{bmatrix}$$

(3 × 9 coefficient matrix)

(5.12a)

The nonlinear coefficient matrix (cm) elements appear now as:

$$W^{\nabla}_{onm} \equiv \langle W_{r_o}(y) I_3 | W_{r_n}(y) \vec{\nabla} W_{r_m}(y) \rangle$$

$$= \int_D W_{r_o}(y) W_{r_n}(y) (3 \times 9 \text{ cm})\, dy \qquad (5.13)$$

With the introduced abbreviations the system (5.12) may be written as:

$$\sum_{r_n \in S} W_{on} \frac{\partial}{\partial t} \vec{V}_1(r_n)$$

$$= \sum_{r_n \in S} \left(-W^{U\nabla}_{on} \vec{V}_1(r_n) - W^{\nabla:U}_{on} \vec{V}_1(r_n) + \nu W_{on} \vec{V}_1(r_n) - \sum_{r_m \in S} W^{\nabla}_{onm} \vec{V}_1(r_n) \vec{V}_1(r_m) \right)$$

$$= \sum_{r_n \in S} \left[-W^{U\nabla}_{on} - W^{\nabla:U}_{on} + \nu W^{\Delta}_{on} - \sum_{r_m \in S} W^{\nabla}_{onm} \vec{V}_1(r_m) \right] \vec{V}_1(r_n); \; r_o \in S \qquad (5.14)$$

As can be easily seen by comparing the nonlinear term in Equation (5.14) with the related term in Equation (4.7) (using the definition of the advective operator (4.3)), the discretization performs the following transition:

$$\vec{V}_1[\vec{\nabla} : \vec{V}_1] \xrightarrow{\text{discretization}} \sum_{r_n, r_m \in S} W^{\nabla}_{onm} \vec{V}_1(r_n) \vec{V}_1(r_m); \quad r_o \in S \quad (5.15)$$

The discretization procedure shows that this transition can be generalized directly to:

$$\vec{V}_i[\vec{\nabla} : \vec{V}_j] \xrightarrow{\text{discretization}} \sum_{r_n, r_m \in S} W^{\nabla}_{onm} \vec{V}_i(r_n) \vec{V}_j(r_m); \; r_o \in S \quad (5.16)$$

or:

$$\vec{V}_j[\vec{\nabla} : \vec{V}_i] \xrightarrow{\text{discretization}} \sum_{r_n, r_m \in S} W^{\nabla}_{omn} \vec{V}_i(r_n) \vec{V}_j(r_m); \quad r_o \in S, \qquad (5.17)$$

where the indices n,m in W^{∇}_{onm} change places. This difference between (5.16) and (5.17) follows easily from the definition (5.13).

With the transitions (5.16), (5.17) and the definition of the advection operator (4.3), Equations (4.9) and (4.11) can directly be translated into their discretized approximations:

$$\sum_{r_n \in S} W_{on} \frac{\partial}{\partial t} \vec{V}_2(r_n) = \sum_{r_n \in S} \Big[-W^{U\nabla}_{on} \vec{V}_2(r_n) - W^{\nabla : U}_{on} \vec{V}_2(r_n) + \nu W^{\Delta}_{on} \vec{V}_2(r_n)$$

$$- \sum_{r_m \in S} W^{\nabla}_{onm} \vec{V}_2(r_n) \vec{V}_2(r_m) - \sum_{r_m \in S} (W^{\nabla}_{onm} + W^{\nabla}_{omn}) \vec{V}_1(r_n) \vec{V}_2(r_m) \Big];$$

$$r_o \in S \qquad (5.18)$$

$$\sum_{r_n \in S} W_{on} \frac{\partial}{\partial t} \vec{V}_3(r_n) = \sum_{r_n \in S} \Big[-W^{U\nabla}_{on} \vec{V}_3(r_n) - W^{\nabla : U}_{on} \vec{V}_3(r_n) + \nu W^{\Delta}_{on} \vec{V}_3(r_n)$$

$$- \sum_{r_m \in S} W^{\nabla}_{onm} \vec{V}_3(r_n) \vec{V}_3(r_m)$$

$$- \sum_{r_m \in S} (W^{\nabla}_{onm} + W^{\nabla}_{omn}) (\vec{V}_1(r_n) + \vec{V}_2(r_n)) \vec{V}_3(r_m) \Big];$$

$$r_o \in S \qquad (5.19)$$

Finally, the coefficient matrix on the left hand side of Equations (5.14), (5.18), (5.19) is inverted over to the right hand side, as is shown here for Equation (5.14):

$$\frac{\partial}{\partial t} \vec{V}_1(r_n) = \sum_{r_n \in S} W^{-I}_{on} \Big[-W^{U\nabla}_{on} - W^{\nabla : U}_{on} + \nu W^{\Delta}_{on} - \sum_{r_m \in S} W^{\nabla}_{onm} \vec{V}_1(r_m) \Big] \vec{V}_1(r_n);$$

$$r_o \in S, \qquad (5.20)$$

with W_{on}^{-I} denoting the inverse from the left-hand side.

This system (5.20) (relating to Equation (4.7)) is the final result of the discretization procedure. Once the mean flow $\vec{U}$ and appropriate basis functions $W_r(y)$ are selected and thereby the coefficient matrices determined, it can be analyzed for bifurcations and numerical solutions. Therefore, the construction of basis functions determining incompressibility and boundary conditions follows next.

6. DIVERGENGE-FREE BASIS FUNCTIONS

Using Equation (5.8), a velocity field $\vec{V}(r)$ over D approximates like:

$$\vec{V}(r) \approx \sum_{r_n \epsilon S} W_{r_n}(r)\vec{V}(r_n) = \vec{V}_n(r) \tag{6.1}$$

The basis functions $W_{r_n}(r);\ r_n \in S;\ r \in D$ have to have the properties of weight functions, i.e.:

$$W_{r_n}(r_n) = \sup_{r \epsilon D} W_{r_n}(r) = 1 \tag{6.2}$$

$$W_{r_n}(r_m) = 0; \quad r_n, r_m \in S; \quad m \neq n \tag{6.3}$$

The condition (6.3) suggests restricting the support of $W_{r_n}(r)$ within a neighbourhood of r_n.

Although a rectangular discretization is in principle possible and seems to be simpler to implement (since the $r_n \in S$ would sit on a rectangular grid naturally induced by the cartesian coordinate system as well as rectangular areas of supports for the $W_{r_n}(r)$), it leads to complications with the overlapping properties of the supports and condition (6.3). Most applications of finite element techniques therefore discretize the fluid domain into simplices (i.e. triangles in two dimensions and tetrahedrons in three dimensions) with the set S of nodes consisting of all centroids on the interfaces between the simplices. The area of support for $W_{r_n}(r)$ is in this case the union of the two simplices sharing the interface with r_n as centroid. See Figure 11. In this discretization, the individual tetrahedron is the smallest volume element permitted. This means for the incompressibility condition that no net flux is permitted through the faces F enclosing the tetrahedron:

$$\int_F \vec{V}(r)\hat{n}_F dF = 0 \tag{6.4}$$

$\hat{n}_F$ is the unit vector perpendicular to the faces F. Using Greens Theorem and the approximation (6.1) gives:

$$\begin{aligned}\int_F \vec{V}_n(r)\hat{n}_F dF &= \int_T \operatorname{div} \vec{V}_n(r) dr_1 dr_2 dr_3 \\ &= \int_T \sum_i^3 \frac{\partial}{\partial r_i}\left[\sum_{r_n \in S} W_{r_n}(r)\vec{V}(r_n)\right] dr_1 dr_2 dr_3 \\ &= \sum_i^3 \sum_{r_n \in S} \vec{V}(r_n) \int_T \frac{\partial W_{r_n}(r)}{\partial r_i} dr_1 dr_2 dr_3 \end{aligned} \tag{6.5}$$

This has to vanish for all possible $\vec{V}(r_n)$; $r_n \in S$. Therefore:

$$\int_T \frac{\partial W_{r_n}(r)}{\partial r_i} dr_1 dr_2 dr_3 = 0; \quad i = 1,2,3 \tag{6.6}$$

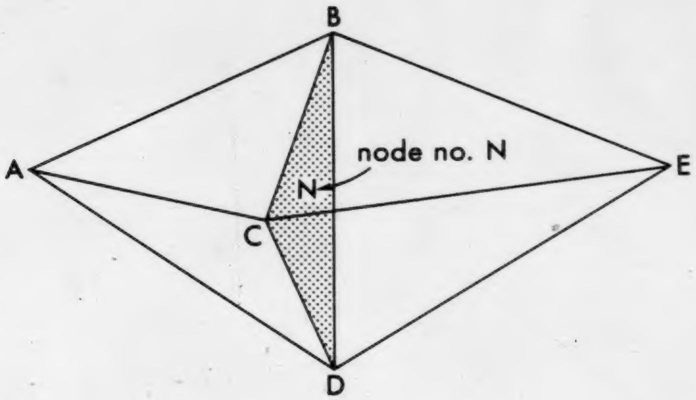

B
node no. N
A
N
E
C
D

FIGURE 11. Two Tetrahedrons Sharing a Face with Centroid (Support for Finite Elements)

It has been shown by Fortin [11] that only piecewise continuous basis functions or so-called non-conforming finite elements are compatible with the incompressibility condition. The simplest possible structure for the $W_{r_n}(r)$ are therefore polynomials defined over the two support tetradehrons individually with $W_{r_n}(r)$ vanishing over the rest of D. That means, the $W_{r_n}(t)$ will be discontinuous over the faces of their support tetrahedrons. Conditions (6.2), (6.3), (6.6) determine alltogether ten conditions per tetrahedron and polynomial since the condition $W_{r_n}(r_n) = \sup_{r \in D} W_{r_n}(r)$ can be written as:

$$\text{grad } W_{r_n}(r_n) = 0$$

$$\text{or:} \quad \frac{\partial W_{r_n}(r_n)}{\partial r_i} = 0, \quad i = 1,2,3 \tag{6.7}$$

This suggests the following polynomial for $W_{r_n}(r)$:

$$W_{r_n}(t) = C + \sum_{i}^{3} C_i r_i + \sum_{i \leq j}^{3} C_{ij} r_i r_j \tag{6.8}$$

Now a linear algebraic system of ten equations can be set up to solve for the ten coefficients C, C_i, C_{ij} of Equation (6.8). This can be considerably simplified by the introduction of barycentric coordinates (see Appendix B) λ_L; $L = 1,2,3,4$:

$$r_i = \sum_{L}^{4} r_{iL} \lambda_L \tag{6.9}$$

$$1 = \sum_{L}^{4} \lambda_L \tag{6.10}$$

Here r_{iL} is the i-component of the L-th vertex in cartesian coordinates. A derivation of this 10 × 10 system can be found in the Appendix B.

7. DIRICHLET BOUNDARY CONDITIONS AND STRUCTURE OF MEAN FLOW $\vec{U}$

As described in the Appendix B, the discretization program partitions a given rectangular domain D in fluid space into rectangular elements, each one of which is subdivided into five tetrahedrons (see Figures 12a,b). Omitting all nodal points on the surface planes ∂D of D from the discretization, i.e., setting their basis functions or finite elements identical to zero:

$$W_{r_{\partial D}}(r) = 0 \tag{7.1}$$

means Dirichlet boundary conditions for the perturbation approximations $\vec{V}_n$:

$$\vec{V}_n(r_{\partial D}, t) = 0 \quad \forall t, \tag{7.2}$$

as follows directly from the approximation (6.1). This is equivalent to setting the $\vec{V}(r_{\partial D})$ in (6.1) equal to zero. It is evident that, due to the construction of the discretized ODE-systems, these Dirichlet boundary conditions hold for the perturbations $\vec{V}_1$, $\vec{V}_2$, $\vec{V}_3$ only (i.e., to their discretization $\vec{V}_n$ respectively), and not for the main flow $\vec{U}$, which can go through D or ∂D respectively unaffected. This

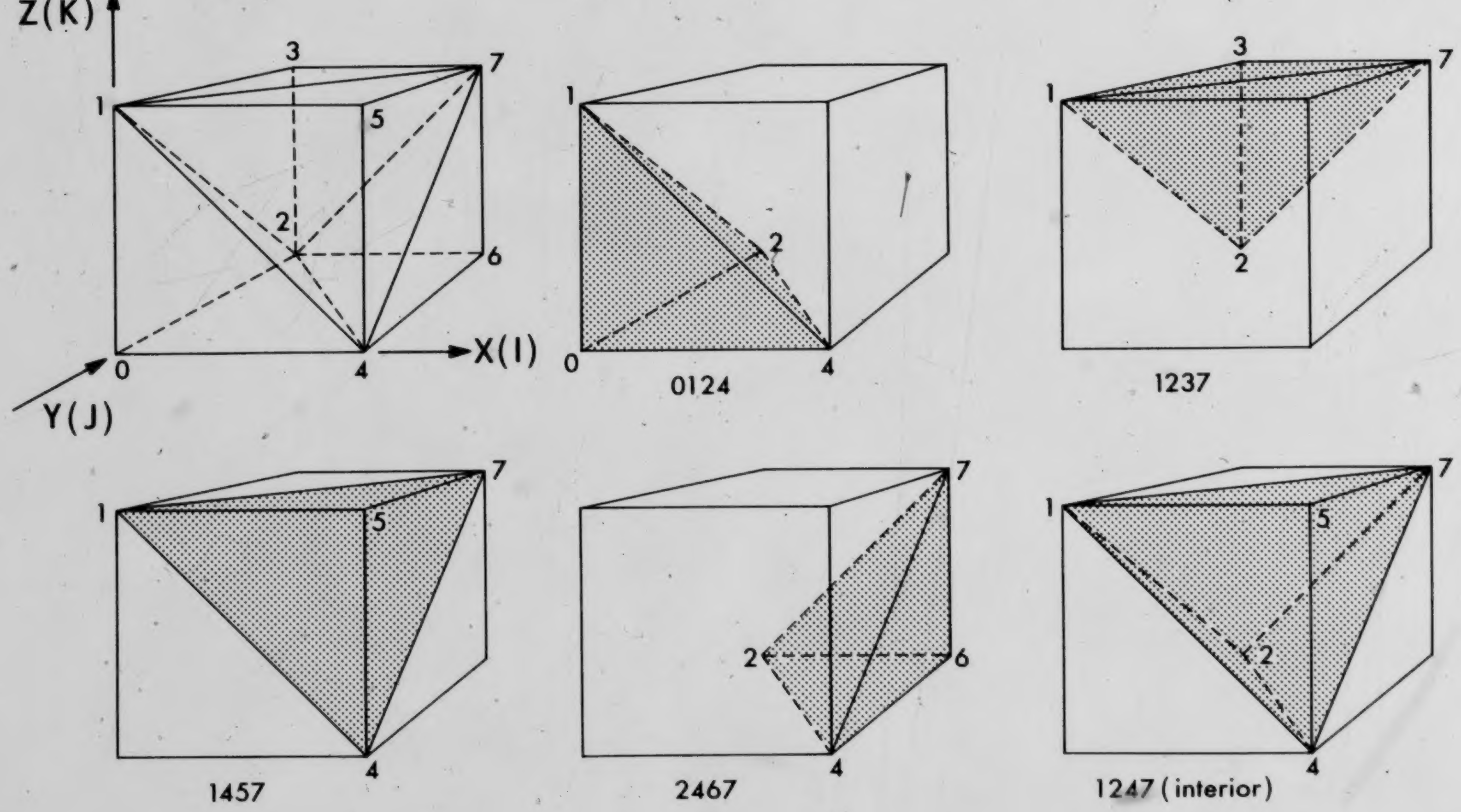

Z(K)
Y(J)
X(I)
0
1
2
3
4
5
6
7
0124
1237
1457
2467
1247 (interior)

FIGURE 12A. A-Configuration of Discretization

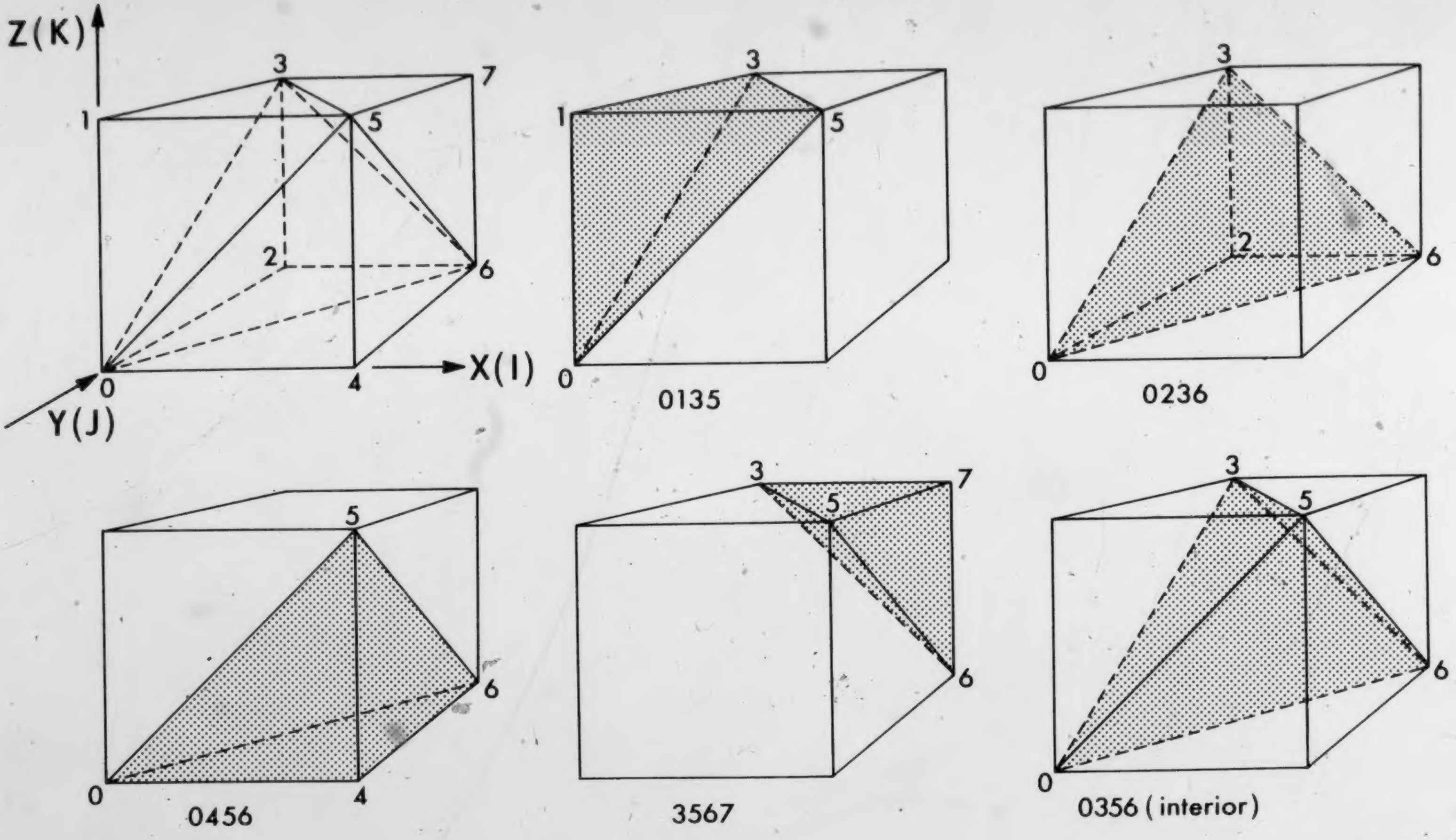

Z(K)
X(I)
Y(J)
0
1
2
3
4
5
6
7
0135
0236
0456
3567
0356 (interior)

FIGURE 12B. B-Configuration of Discretization

creates the following physical situation (see Figure 13): Within a stationary flow field $\vec{U}_{ext}$, say, which can be a solution of Equation (4.1) or (4.4), respectively, there exists a rectangular domain D within which $\vec{U}_{ext}$ can be approximated by $\vec{U}$. The stability properties of $\vec{U}$ within D, i.e. the development of a perturbation $\vec{V}_1$ on top of $\vec{U}$, is determined by Equation (4.7) or its discretization (5.20) respectively. However, due to the Dirichlet boundary conditions, this perturbation $\vec{V}_1$ cannot propagate out of D.

For $\vec{U}$, the following structure is built into the discretization program:*

$$\begin{aligned} U_1 &= C_1 + C_{11}X_1^{E_{11}} + C_{12}X_2^{E_{12}} + C_{13}X_3^{E_{13}} \\ U_2 &= C_2 + C_{21}X_1^{E_{21}} + C_{22}X_2^{E_{22}} + C_{23}X_3^{E_{23}} \qquad (7.3) \\ U_3 &= C_3 + C_{31}X_1^{E_{31}} + C_{32}X_2^{E_{32}} + C_{33}X_3^{E_{33}} \end{aligned}$$

For the exponents, integers between 1 and 3 inclusive have to be chosen as input for the discretization program. The coefficients, however, are real and variable parameters of the system (14.4). Besides the viscosity ν, they can therefore be used as stability-determining parameters for bifurcations or as functions thereof.

---

*In principle, any conceivable structure of $\vec{U}$ can be built into a discretization program for (5.20). The structure (7.3) however is relatively easy to program and offers a considerable choice for bifurcation parameters. See Appendix B for details.

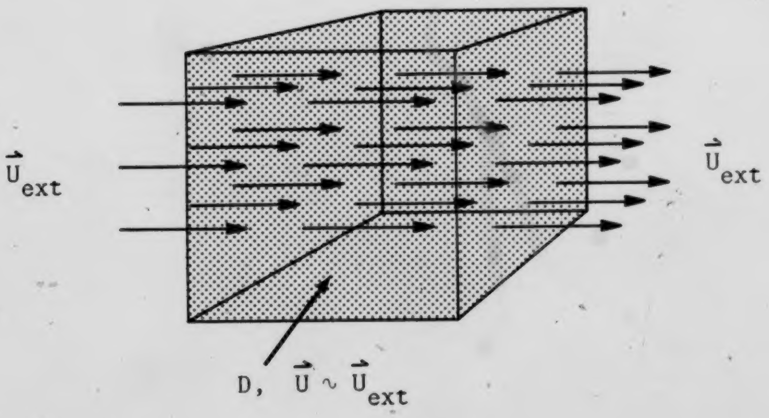

$\vec{U}_{ext}$
$\vec{U}_{ext}$
D, $\vec{U} \sim \vec{U}_{ext}$

FIGURE 13. Flow Through Domain D by $\vec{U}$

PART 3

8. INVESTIGATION OF BIFURCATIONAL PROPERTIES

To test the system (5.14) for Hopf bifurcations, BIFOR 2, a FORTRAN program written by B. Hassard* is used. Since the CPU time of the program is roughly proportional to N^3, with N as the number of dimensions of the ODE-system, the lowest possible discretization had to be used. That is, a domain D consisting of one cube with corner coordinates (0,0,0), (0,0,1), (0,1,0), (1,0,0), (0,1,1), (1,0,1), (1,1,0), (1,1,1) is split into five tetrahedrons according to Figure 12 (A-configuration). Due to Dirichlet boundary conditions, only the 4 nodes on the interior tetrahedral faces contribute to the system (5.14), which therefore becomes 12-dimensional.

Since BIFOR 2 is explained very extensively and sufficiently in theory and function by its author in [5], only a brief introduction and description is given here.

BIFOR 2 varies the bifurcation parameter η in the Jacobian matrix $A(X_*(\eta),\eta)$ of the ODE system until the set of eigenvalues $\lambda_i(\eta)$, $i = 1,\dots,N$, consists of one complex-conjugate pair with zero real parts, and the real parts of the rest of the set are negative. This is performed by a secant iteration. In between each two of these secant iteration steps, the stationary solution $X_*(\eta)$ for the new iterate of η is determined by Newton iteration before the new set of eigenvalues is evaluated. Since the stationary solution of the system (5.14) of interest is always the trivial solution $\vec{V}_1(\eta) \equiv 0$, $\forall\eta$ the

---

[*] Department of Mathematics, SUNY, Buffalo, New York.

user-supplied initial guess for the stationary solution at the critical value η_c of the bifurcation parameter is already known to be $\vec{V}_1(\eta_c) \equiv 0$. Once $X_*(\eta_c)$ and η_c are found, the program calculates all parameters to be used in the following three equations:

$$X(t,\eta) = X_*(\eta_c) + \left[\frac{\eta-\eta_c}{\mu_2}\right]^{\frac{1}{2}} \mathrm{Re}\left(e^{2\pi it/T(\eta)}\vec{V}_{ec}\right) \tag{8.1}$$

$$T_p(\eta) = \frac{2\pi}{\omega_o}\left[1 + \tau_2\left[\frac{\eta-\eta_c}{\mu_2}\right]\right] \tag{8.2}$$

$$\beta(\eta) = \beta_2\left[\frac{\eta-\eta_c}{\mu_2}\right] \tag{8.3}$$

These are first-order approximations for the periodic perturbation $X(t,\eta)$, its period $T(\eta)$, and the Floquet exponent $\beta(\eta)$, which determines the stability of $X(t,\eta)$:

$$\begin{aligned} \beta(\eta) > 0 &: X(t,\eta) \text{ is unstable} \\ \beta(\eta) < 0 &: X(t,\eta) \text{ is stable} \end{aligned} \tag{8.4}$$

μ_2 is a coefficient from a Taylor expansion. It determines the criticality of the bifurcation:

$\mu_2 > 0$: supercritical bifurcation; $X(t,\eta)$ exists on RHS of η_c
$\mu_2 < 0$: subcritical bifurcation; $X(t,\eta)$ exists on LHS of η_c

$\vec{V}_{ec}$ is the complex eigenvector of the Jacobian matrix $A(X_*(\eta_c),\eta_c)$ and ω_o is the imaginary part of the complex conjugate pair of eigenvalues with zero real part at $\eta = \eta_c$. τ_2 and β_2 are coefficients from Taylor expansions; they and μ_2 are determined as analytic functions of the first coefficient of the Poincaré normal form of the ODE system.

$$\dot{\xi} = \lambda(\eta)\xi + \sum_{j=1}^{L/2} C_j(\eta)\xi|\xi|^{2j} + O(|\xi|\,|(\xi,\eta)|^{L+1}) \qquad (8.5)$$

This form is obtained by a coordinate transformation of the system and τ_2, β_2, μ_2 are given by:

$$\mu_2 = \frac{-\mathrm{Re}(C_1(0))}{\alpha'(\eta_c)} \; ; \qquad \beta_2 = 2\mathrm{Re}(C_1(0)) \qquad (8.6)$$

$$\tau_2 = \frac{-(\mathrm{Im})(C_1(0)) + \mu_2\omega'(\eta_c))}{\omega_o} \qquad (8.7)$$

with the first coefficient of (8):

$$C_1(0) = \frac{i}{2\omega_o}\left[(2U_1^T f_{20})(2U_1^T f_{11}) - 2|2U_1^T f_{11}|^2 - \frac{1}{3}|2U_1^T \bar{f}_{20}|^2\right] + \frac{G_{21}}{2} \qquad (8.8)$$

here the U_1^T are transposes of the left eigenvector of the Jacobian matrix $A(X_*(\eta_c),\eta_c)$ for the eigenvalue $\lambda_1(\eta_c) = i\omega_o$ and f_{20}, f_{11}, G_{21} are partial derivatives of the right-hand side of the ODE system. Moreover:

$$\lambda_1'(\eta_c) = \alpha'(\eta_c) + i\omega'(\eta_c) \qquad (8.9)$$

is the eigenvalue $\lambda_1(\eta)$ derived after η at η_c. The program evaluates these equations* and thereby supplies the numerical values for all the parameters in Equations (8.1), (8.2), (8.3). Their complete derivation can be found in [5].

---

*Except for the Poincaré normal form (8.5), since only its coefficient $C_1(0)$ is used.

The first flow to be tested was:

$$U_x = 0; \quad U_y = C_{yx}X; \quad U_z = 0 \tag{8.10}$$

with : $\eta = C_{yx}$

Around 80 runs were performed and for bifurcation parameters the coefficient C_x or C_{yx} were used. After initial trial runs, it turned out that the flow was unstable for positive viscosities throughout. That is, BIFOR 2 could not zero any of the real parts of the eigenvalues; they were all positive and bounded away from zero. Since this behaviour was observed for all flows tested so far, negative viscosities had to be used throughout. For the flow (8.10) viscosity values between -.01 and -5 were used. For a viscosity of -1, the flow became unstable at $C_{yx} \approx 6.56$. However, the bifurcation was not Hopf and the perturbation $\vec{V}_1$ therefore stationary. This was found for all bifurcations for the flow (8.10). Moreover, it turned out for this and all other tested flows that the critical value for the bifurcation parameter is proportional to the absolute value of the viscosity:

$$\eta_c \sim |\nu| \tag{8.11}$$

The phenomenon of stationary bifurcation shows that the flow (8.10) somehow changes its structure before it can undergo a Hopf bifurcation. Therefore, nonzero initial guesses for the perturbation $\vec{V}_1$ were experimented with to find a Hopf bifurcation for a flow of the type (8.10) + $\vec{V}_1(\eta_c)$. However, the program always either converged to the trivial solution $\vec{V}_1 \equiv 0$ or error-terminated in the Newton iterations. Moreover, the additional Newton iterations increased the CPU time drastically and this approach was therefore abandoned for all other tested flows as well.

A listing of all the other tested flows with a discussion of their bifurcational properties is given in Appendix C. Here, only the main characteristics are summarized:

1. As mentioned already, only negative viscosities can be used. All tested flows seem to be unstable for positive viscosities.*
2. Flows with linear or no shear or one component only, have only real eigenvalues. As a rule, the program can find the critical value of the bifurcation parameter, where the largest eigenvalue vanishes. This can be understood as an inability of no shear or linear shear to develop periodic perturbations. These types of profiles have to reset themselves (e.g., by a stationary bifurcation) to nonlinear ones before they can undergo a Hopf bifurcation.
3. The critical value η_c and the frequency ω_0 are proportional to the magnitude of the viscosity, whereas β_2 and μ_2 are inversely proportional.
4. Most of the flows show qualitatively correct behaviour; an increase of shear in the mean flow $\vec{U}$ leads to its instability. In the case of linear shear this occurs by a stationary bifurcation and

---

*See Appendix C for an exception.

in the case of nonlinear shear a supercritical stable Hopf bifurcation occurs.

To obtain a superposition of three frequencies leading to a 3-torus as the invariant manifold in solution space, the following idea was applied first.

The analytic approximation for $\vec{V}_1(r,t)$ according to Equation (8.1) is evaluated for a set of fixed time points t_i, $i = 1,\ldots,N$, say, and the obtained values $\vec{V}_1(r,t_i)$ are substituted into the system for $\vec{V}_2(r,t)$ (i.e., Equation (5.18)) and tested for bifurcations one after the other. This way one obtains N secondary orbits for the second periodic perturbation $\vec{V}_{2i}(r,t)$, $i = 1,\ldots,N$, again by Equation (8.1). That is, the 2-torus relating to two bifurcations is discretized into N closed orbits. Substituting the $\vec{V}_1(r,t_i)$ and evaluations at fixed time points for the secondary orbits $\vec{V}_{2i}(r,t_j)$, $j = 1,\ldots,M$, say, into the system for $\vec{V}_3(r,t)$ (i.e., Equation (5.19)) and testing for bifurcations one obtains $N \times M$ orbits for the third bifurcation, i.e., each secondary orbit is discretized into M points which degenerate into orbits for the third bifurcation. This $N \times M$ discretization of the 3-torus can now be used to approximate any trajectory on it by use of interpolation methods. However, this approach was abandoned since for the flows tested secondary and third order bifurcations could not be obtained for each different time setting on the first orbit and secondary orbits. Moreover, it is not established yet whether this type of discretization in time is legitimate for the testing of non-autonomous systems for higher-order bifurcations. On the other hand, the flows which showed such a sequence of a first, secondary and third order bifurcation at least for some time settings turned out to have better recurrency properties, as was found out by the second approach, which combines BIFOR 2 with ODE solvers.

9. APPLICATION OF ODE SOLVERS

Since the construction of a discretized torus cannot be pursued because of the reasons mentioned, numerical solutions generated by ODE solvers offer a reasonable alternative. For this method, the non-autonomous system for $\vec{V}_2(r,t)$ was used with the approximation by Equation (8.1) for $\vec{V}_1(r,t)$ determined by BIFOR 2 .

Both predictor-corrector* and Runge-Kutta** solvers were applied. Unfortunately, the corrector-predictor method had to be abandoned despite the favourable CPU times (which are about one fourth of the Runge-Kutta programs) since it caused in all its options and variations a "sawtoothing" on the solutions, as can be seen by Figures T1, T2 in Appendix F. However, the CPU time on the single-precision RKF45 routine could be dropped to about one sixth of the double-precision DVERK routing by recompiling the user-supplied routine which evaluates the ODE-system on a FORTRAN 77 compiler with highest efficiency option.*

---

*IMSLDPLIB DGEAR: Gears or Adams corrector-predictor.

**IMSLDPLIB DVERK; Sixth order Runge-Kutta-Verner method.

RKF45: Fourth-fifth order Runge-Kutta-Fehlberg method.
Authors: L.F. Shampine, H.A. Watts; Sandia Corporation,
Albuguergue, New Mexico 87115.
See [29] for a description of RKF45.

The following four flows were used for the described application of ODE solvers:

(I) $U_x = 6\eta Z; \quad U_y = 4\eta X^2; \quad U_z = \eta Y^3; \quad \nu = -1.$

This flow bifurcates stable and subcritically at $\eta = .872$ with a period of $T_p = 3.24$.** Higher order bifurcations using the described method of torus discretization could not be found except for a stationary one with $\vec{V}_1(r,t=0)$ and at $\eta \approx .952$. Numerical solutions were attempted at $\eta = .9, .915, .93, 1$ with initial conditions equal to .001 for all twelve velocity components of $\vec{V}_2(r,t=0)$. For $\eta = .9$ the flow was completely stable in its first bifurcation. The running time was $T = 30$ (300 time steps with .1 for step size) and $\vec{V}_2$ vanished completely. For $\eta = .915$, however, $\vec{V}_2(r,t)$ grew unbounded with little sign of recurrency and therefore error-terminated at $T = 6.2$. Essentially the same behaviour was observed for $\eta = .93$ and 1 with an error-termination at $T = 5.3$ and 2.6.

(II) $U_x = 5\eta Y^2; \quad U_y = 3\eta Z^2; \quad U_z = \eta X^2, \quad \nu = -1.$

---

*CPU times on the U/A AMDAHL were by order of magnitude ca. 150 seconds for 1000 timesteps with stepsizes of ca. .02 - .05 simulation time on DVERK IMSLDPLIB for both random perturbed and unperturbed runs. This could be reduced to ca. 60 sec. for random perturbed runs and ca. 25 - 30 sec. for unperturbed runs on RKF45.

**All times T or T_p are understood as simulation time.

The first bifurcation occurs (stable and supercritical) at $\eta = 1.488$ with a period of $T_p = .23$. A secondary Hopf bifurcation occurs at $\eta = 1.881$ for $\vec{V}_1(r,t=0)$. ODE runs were tried at $\eta = 1.9$, 2.23, 2.25, 2.3, 2.4 with $\vec{V}_2(r,t=0) = .0001$ for all twelve components. For $\eta = 1.9$, the primary orbit was still stable despite the secondary bifurcation at $\eta = 1.881$; after $T = 30$, the twelve components of $\vec{V}_2$ were of an order of magnitude of around 10^{-6}. For $\eta = 2.23$, 2.25, 2.3, 2.4, the run error-terminated at $T = 14.42$, 21.25, 11.19, 4.6 due to unbounded growth of $\vec{V}_2$. For none of these parameter settings did $\vec{V}_2$ show any significant periodicity or recurrency, and the unbounded growth causing the error termination occured rather suddenly.

(III)
$$U_x = -9\eta + (6+9\eta)Y^2$$
$$U_y = -6\eta + (5+6\eta)Z^2$$
$$U_z = -\eta + (3+\eta)X^2$$
$$\nu = -1.$$

This flow undergoes the first Hopf bifurcation at $\eta = .189$, the secondary one for $\vec{V}_2(r,t=0)$ at $\eta = .637$ and the third order one for $\vec{V}_3(r,t=0)$ at $\eta = 1.003$. The affine functional dependencies of the shear coefficients on the bifurcation parameter η seem to increase the chances for bifurcations up to third order considerably. Amongst the tested flows with pure linear dependencies, none could be found with bifurcations up to third order.

The tested parameter values for this flow were $\eta = .4$, .5, .52, .53, .55, .6, .7 . The Figures T1, T2 (Appendix F) show projected trajectories for $\vec{V}_1 + \vec{V}_2$ at $\eta = .55$ and a period of $T_p = 1.66$ for $\vec{V}_1$. Strong recurrency is evident, but, as can be concluded by the graphs, unbounded growth occurred which error-terminated the run at $T = 24.523$. The trajectories for all other parameter settings were similar with unbounded growth and error-termination occurring for $\eta = .6$ and $.7$ at $T = 9.4$ and 7.44 respectively. For the settings of η lower than $.55$ unbounded growth did not occur within the chosen time settings, which went as high as $T = 40$ for $\eta = .52$.

(IV)

$$U_x = -9\eta + (4 + 9\eta)Y^2$$

$$U_y = -6\eta + (5 + 6\eta)Z^2$$

$$U_z = -\eta + (3 + \eta)X^2$$

$$\nu = -1.$$

Although the only difference between this flow and flow (III) is in the affinity constant of the shear coefficient in the U_x-component, the trajectories of the solution curve looked somewhat more promising as far as the resemblance to projections of 2- and 3 - tori is concerned. Therefore, it has been tested much more extensively. The flow bifurcates at $\eta = .242$ and at $\eta = .329$ and $.347$ for $\vec{V}_1(r,t = 0)$ and $\vec{V}_2(r,t = 0)$ respectively. ODE runs were performed between $\eta = .28$ and $.5$. All graphs (Appendix F) show projections of the $\vec{V}_1 + \vec{V}_2$ solution curve onto the three velocity component planes for each one of the four monitored points in D.

Following these graphs from lower to higher values of η with initial conditions for $\vec{V}_2$ close to zero, it can be seen that the orbit for $\vec{V}_1$ is being followed closely (i.e., $\vec{V}_2$ stays small) for a number of revolutions which decreases with increasing η; e.g., for $\eta = .3$ (Figure T3) this primary orbit is being closely retraced for ca. 6 times, whereas at $\eta \simeq .45$ (Figure T4) only for ca. 1½ times. After these initial revolutions a number of transitional revolutions follows (again decreasing with increasing η), where the system passes from the $\vec{V}_1$-orbit to its new invariant manifold. For $\eta = .3$ (Figure T3), this invariant manifold might not have been reached within a time $T = 30$. For all settings of $.325 \leq \eta < .449$, (Figures T5, T6, T7, T8) the running time was sufficient to reveal a limit cycle as invariant manifold; except for $\eta \geq .449$ (Figure T4) where the solutions start to grow unbounded. The shape of a double loop of this limit cycle, which can be seen by most of these projections, suggests period doubling. It could however by determined by setting marks on the trajectories after every period $T_p(\eta)$ as calculated by Equation (8.2) that this period stays roughly the same for $\vec{V}_1 + \vec{V}_2$.

For increasing η the limit cycle becomes more attracting. It changes and grows in size up to $\eta = .449$ (Figure T4), where it seems to "fan out" in a particular section due to an increase of the X-component on top of it's periodicity, as can be seen by projections into the XY- and XZ-planes. The running time of $T = .32$ however was not sufficient to determine whether the X-components will grow unbounded or approach a finite limit. Increasing η by .001 to $\eta = .45$* causes an error-termination due to unbounded growth of the X-components at $T = 22.82$. Moreover a spreading of the trajectories can be observed not only in the sections of this error termination, but in other sections as well. To observe some of the behaviour of the flow for $\eta \geq .45$ (Figure T9), after an error termination due to unbounded growth, the last determined value of $\vec{V}_2$ was reduced by multiplying it with a factor in the orders of magnitude of 10^{-4} to 10^{-6} to continue the run. Evidently the legitimacy of this is questionable, since beyond $\eta = .45$, the flow seems to be in a region where it cannot be described any more by the present model. The assumption for this reduction trick is that the model still works reasonably well close to the time point of sudden unbounded growth even for $\eta \geq .45$ in case of the present flow, and it can therefore be understood as a sequence of different runs with different initial conditions. Therefore, the graphs for $\eta = .47$ (Figure T9) show wider spacing of trajectories and a certain randomizing effect due to this reduction method is evident too. Moreoever, Y- and Z-components can grow unbounded as well for these higher parameter settings, as the YZ-projection in particular shows.

---

*See Figure T4 ($\eta = .449$), which looks identical (except for unbounded growth).

In case of the present "one cube discretization" the coordinates of the four points in fluid space are:

Point 1	(1/3, 1/3, 1/3)
Point 2	(1/3, 2/3, 2/3)
Point 3	(2/3, 1/3, 2/3)
Point 4	(2/3, 2/3, 1/3)

according to the A-configuration (see Appendix). By the amplitude (i.e., "size") and shape of the trajectories of their associated velocity vectors, these four points fall into two groups. Group one, consisting of point 1 and 3 has $\vec{V}_1$ orbits which are ca. one to two orders of magnitude larger than those of group two, i.e., point

2 and 4. This seems to be the reason why the limit cycles of $\vec{V}_1(r = \text{Point } 1,t) + \vec{V}_2(r = \text{Point } 1,t)$ and $\vec{V}_1(r = \text{Point } 3,t) + \vec{V}_2(r = \text{Point } 3,t)$ stay within the order of magnitude of the orbits of $\vec{V}_1(r = \text{Point } 1,t)$ and $\vec{V}_1(r = \text{Point } 3,t)$ respectively up to the occurrence of global instability at $\eta = .449$ (Figure T4) whereas the ratio of these respective orbits and limit cycles for group two is at least one order of magnitude. For group two all six projections show the double loop in the limit cycle whereas for group one it can be seen only in the XY projection of point one and the YZ projection of point three.

10. SIMULATION OF RANDOM PERTURBATIONS AND THEIR EFFECT (ON FLOW IV)

To simulate the aforementioned "appropriate" combination of perturbations in connection with the use of ODE-solvers, IMSLLIB random number generators are being used on Flow (IV). The four points are being perturbed independently of each other in the following way: Routine GGSPH supplies the coordinates of a point on the unit sphere random determined under the assumption of uniform probability distribution over the sphere. These coordinates are being multiplied with one and the same factor, which is random generated by the routine GGNQF, a generator of Gaussian distributed variates with mean zero and a user-determined variance V_s. The three component perturbations obtained this way are being added to the appropriate velocity components of the point considered perturbed. After that, GGNQF is used again to random-generate the number of unperturbed timesteps to follow. The average of this number, i.e., the variance V_t, is again, as for V_s, set by the user. This time, only the absolute integer part of the generated number is used, since timesteps are positive and integer only. That is, the number of perturbations decreases with increase of the temporal variance. The main characteristic of the system under random perturbations is a global instability for all tested bifurcation parameter values. For a setting of $\eta = .32$ (Figure T11) and a spatial variance of as low as $V_s = .005$ with a temporal variance of $V_t = 2$, most of the projections still showed a strong resemblance to the ones without random perturbations at $\eta = .32$ (Figure T10). The run, however, error-terminated due to unbounded growth at $T = 11.35$. On the other hand, a run of the same bifurcation parameter setting and temporal variance and $V_s = .03$ showed a rather high degree of randomness especially for the projections of the points of group two but would not error-terminate within the total running time of $T = 15$. However, on the basis of the runs performed so far probability assumptions about stability of a certain combination of bifurcation parameter and random variances still seem to be reasonable. That is for lower η and V_s and higher V_t, the chances for a "more stable" behaviour are better. To overcome this problem of global instability, the reduction method as described earlier was again employed. (In general, the projections of group two show a higher degree of randomness than group one.)

Although the spatial structure of the random-perturbed trajectories is in most cases quite similar to the unperturbed trajectories, their temporal structure is quite different and shows more randomness. For example, there are regions where sets of trajectory segments are organized so that they appear nearly parallel to each other. However, these sections are not being traced in some consecutive order, but rather randomly, as can be seen when the trajectory is being drawn on a graphics device.

11. SPECTRAL ANALYSIS (OF FLOW IV)

To obtain a comparison with the inertial subrange law (Appendix E), fast Fourier transforms were performed both on most of the random-perturbed ODE-runs as well as on the unperturbed ODE-runs.

First, the magnitudes $\vec{V}(r_j,t_i)$ of the total velocity vectors for each timestep t_i and each of the four monitored points r_j, $j = 1,\ldots,4$ are determined by the (obvious) formula:

$$|\vec{V}(r_j,t_i)| = \left[\sum_{k=1}^{3} (V_{\text{stat }k}(r_j) + V_{1k}(r_j,t_i) + V_{2k}(r_j,t_i))^2\right]^{\frac{1}{2}} \quad (11.1)$$

with k counting through the X,Y,Z-components and V_{stat} = velocity from steady input profile; $\vec{V}_1$ = first perturbation from bifurcation; $\vec{V}_2$ = second perturbation growing on $\vec{V}_1$ and determined by ODE-solvers.

After that, the mean

$$\overline{|\vec{V}(r_j)|} = \left[\sum_{i}^{i_{\max}} |\vec{V}(r_j,t_i)|\right] \frac{1}{i_{\max}} \quad (11.2)$$

taken over all $i_{\max}$ timesteps is subtracted from each individual $|\vec{V}(r_j,t_i)|$, to obtain a pure time-dependent part $\vec{V}_p$ of the velocity:

$$|\vec{V}_p(r_j,t_i)| \equiv V_p(r_j,t_i) = |\vec{V}(r_j,t_i)| - \overline{|\vec{V}(r_j)|} \quad (11.3)$$

This way, a possible nonzero mean of $\vec{V}_2$ stemming from its nonsymmetric trajectory w.r.t. zero is also subtracted.

The velocities $\vec{V}_p$ are now analyzed for their power spectra in two pairs of two points (1,2 and 3,4) by the IMSLLIB routine FTFPS to obtain frequency power spectra and cross spectra estimates. For this spectral analysis, the entire time series, i.e., a particular ODE-run in this case, is subdivided into intervals of equal size, which must be a power of two, and whose number must divide the number of samples in the entire time series evenly. If this is not possible, the complete number in the time series is increased by samples of value zero, commonly referred to as "padding". For example, in the case of 1000 timesteps, which was the largest and most frequently used number of timesteps, subinterval sizes of 64 (= 2^6) or 128 (= 2^7) were chosen. A time series of size 1024, contains exactly 16 or 8 respectively of these subsamples. A run of 1000 timesteps was therefore padded with an extra 24 timesteps of zero value.

Each one of these subsamples is Fourier-analyzed and an average spectrum is determined from all these subsample spectra. Clearly, this causes a smoothing in this average spectrum, which is necessary to determine a definite slope or trend, that would be obscured by dense oscillations in a spectrum obtained without averaging.

To obtain a direct comparison with the inertial subrange law (Appendix E) the spectrum frequencies are rescaled as wavenumbers under the assumption of the Taylor hypothesis. The diagrams show the power spectra (i.e., spectral energy density) on the ordinate versus wavenumbers on the abscissa, with both axes on decadic log scale.

A spectral analysis was performed for the following eleven parameter settings:* η = .27(p) (T12,S12), .28(p) (T13,S13), .3(6) (T3,S3,T14,S14), .325(b) (T5,S5,T15,S15), .33(p) (T16,S16),

---

*See Appendix F for Figures; T (trajectories), S (spectra).
p = random perturbations superimposed.
U = without random perturbations.
b = both with and without random perturbations.

.35(U) (T6,S6), .4(U) (T7,S7), .445(p) (T17,S17), .447(U) (T8,S8), .449(U) (T4,S4), .47(U) (T9,S9). For lower parameter settings, such as η = .27,.28, the spectra show an almost horizontal section for wavenumbers around .2 to .5 and spectral densities around .3 for group I (points 1,3) and wavenumbers around .2 to .7 with spectral densities around .07 to .001 for group 2 (points 2,4). This difference of ca. one order of magnitude between the spectra of the two groups reflects the already mentioned difference in size of one order of magnitude between the limit cycles of the two groups. This section in the spectrum can clearly be associated with the production range of large eddies in a turbulent flow.

After this initial section, the spectra fall off with a slope of roughly -7 to wavenumbers of ca. 2 to 3 and spectral densities around 10^{-4} to 10^{-5}. The ranges between this section and wavenumbers of ca. 8 to 9 shows more of an oscillatory behaviour in their spectra, which is due to the logarithmic scale and therefore a higher density of data points on the graph. However, the mean slope in these sections compares quite well to -5/3.

Increasing η to .3 changes the slope of the midsection from ca. -7 to ca. -5 and a slight steepening of the mean slope in the right section for large wavenumbers from ca. -5/3 to ca. -5/2. This spectrum stays essentially the same for η = .325. Also, the unperturbed spectra for η = .3, .325 show in principle the same shape as their perturbed counterparts just described.

The slopes closest to -5/3 can be found for the spectra of group 1 at η = .33 and .445. An analysis of these spectral slopes is displayed in Table 1 together with the ones of Group 2. In general, Group 2 again has lower spectral densities than Group 1. The concavity of Group 2 at the lower spectral end indicates an eddy production range. Although point 1 shows an overall average slope of -1.7 for $\eta = .33$ and the same in the lower wavenumber range for $\eta = .445$, it must be noted that, for definite answers w.r.t. acceptance of these results, they would have to be reproduced with sample sizes of at least an order of magnitude larger than the presently analyzed ones. Nevertheless, the change in shape and magnitude of the spectra between $\eta = .325$ and .33 is remarkable, especially since it cannot be concluded from the trajectory graphs. The Table 1 clearly shows for the points of Group 1 spectral slopes in general closer to -5/3 than the points of Group 2. This could have something to do with the Group 2 points being in a region of shear high enough so that generation of homogeneous and isotropic turbulence cannot take place.

Looking at the unperturbed spectra for $\eta \geq .35$, the most evident difference w.r.t. the already discussed spectra are the strong oscillations spanning ca. 2 orders of magnitude. These can be expected on theoretical grounds, since the associated trajectories are periodic and therefore, only a discrete set of frequencies or wavenumbers respectively can occur. The spectra for $\eta = .35$ and .4 look essentially the same. The drastic decrease of oscillations for Group 1 and complete disappearance of spectrum for Group 2 at higher wavenumbers (around 1 to 5) indicates the absence of high frequencies, which can be concluded from the associated trajectory graphs. At $\eta = .447$ and .449 the spectra compare again quite well. Compared to the ones at $\eta = .35$

TABLE 1

ANALYSIS OF SPECTRAL SLOPES FOR $\eta = .33$ (S16) AND .445 (S17)

η		.33	.445
Number of Timesteps; Stepsize		445, .03	500, .03
Mean Magnitude of Random Perturbations (V_s)		.01	.01
Mean Number of Unperturbed Consecutive Timesteps (V_t)		2	2
Sample Size/Subsample Size = Number of Subsamples		448/64 = 7	512/64 = 8
Mean Slopes (with their respective wavenumber ranges)	Point 1	-2.6(.3 - 1.5); - .8 (1.5 - 10)	-1.7(.3 - 1.5); - 1 (1.5 - 6)
	Point 2	.5 (.3 - 1); -1.3(1 - 5)	-1.1(1 - 5)
	Point 3	- 2 (.2 - 1); - .7(1 - 4)	-1.5(.2 - 1); - .7(1 - 6)
	Point 4	(see graph)	-2.5(.3 - .6) - 2 (6 - 1)

and .4, the oscillations are denser and span a wider range of spectral density, which has its maximum with close to 8 orders of magnitude for points 3 at $\eta = .447$. In general, a shift to higher spectral densities can be observed, which is a natural consequence of the increase in amplitudes of $\vec{V}_1$ and $\vec{V}_2$ with increasing η. The appearance of strong spectral oscillations for Point 1 in the wavenumber range between 2 and 6 is also conspicuous, since it cannot be interpreted from any particular structure in the trajectory graphs.

A remarkable flattening of mean spectral slope and decrease in amplitudes of spectral oscillations can be seen at the highest parameter setting analyzed, i.e. at $\eta = .47$. The associated trajectory actually consists of 10 short trajectories with an average of 100 timesteps each. Here, the already discussed reduction technique was employed to deal with the unbounded growth of $\vec{V}_2$. This could explain the flattening of spectral slope, which covers only up to ca. one and a half orders of magnitude in spectral density. It is possible that in this range of the parameter η the model loses its applicability due to unrealistically high shears in the fluid, which cannot be modelled on the basis of NS-equations, or at least not by the finite element technique here employed and rather coarse discretization.

12. SUMMARY AND CONCLUSIONS

To begin with the theoretical mechanism of the model, its main characteristic probably lies in the fact that the NS-equations can be considered as being both valid and invalid within its framework. Locally, they are valid, since the system follows sections of NS-solutions within a certain neighbourhood of every point in state space.

Globally, they are not valid, since the perturbed trajectory through state space of the system cannot be considered a NS-solution, if monitored over a time period long enough, since the system transfers between sections, which are not connected. The trajectory of the system is therefore not identical to a solution curve of the NS-equation. This distinction between local and global applicability is connected to a distinction between two types of vectors (or spaces, within which these vectors are defined) upon which part of the model is based.

In any physical context, a vector has three cartesian components in its general form, unless the coordinate system can be rotated and these three components are reduced in number in adapting to the particular geometry at hand. The most important characteristic of these three components is that they are of the same type, e.g., velocity, acceleration or force. Moreover, they are all associated with one and the same point in physical three-dimensional space. Vectors of this type could be considered "physical" vectors, and they are local, since they apply to only one particular point of a system under consideration.

On the other hand, any point in any axis diagram can be considered a vector. If the two axes on the diagram represent different physical quantities, any rotation (except by multiples of $\pi/2$) will lead to a new diagram with its new axes as linear combinations of the old ones; and the new axes cannot be considered physical vectors, because their components (i.e. the old axes) are not of equal physical type. Clearly, the same goes for every point in such an axis diagram, except for points directly on one of the axes. Such a vector could be considered of "mathematical" type, since it may not represent one particular physical quantity at one particular point in physical space; instead it may represent several different physical quantities (two, if the axis diagram is two-dimensional) not necessarily associated with one and the same physical point. If the number of axes (i.e., dimensionality) on the "diagram" is the same as the number of physical quantities involved in the system, one point or vector in the diagram will describe the state of the whole system, i.e., such a vector is global.

In the present case, the mathematical vector is represented by a physical velocity vector field, i.e., its components are physical vectors of one and the same physical type (i.e. velocity), but associated with different points in physical space.

A small random perturbation will now act locally, affecting a certain point in physical space and its neighbourhood and the physical vectors associated therewith. That is, such a perturbation can affect a particular subset of components of a mathematical vector, but it (almost) always will affect all components of a physical vector at once. Such a perturbation will momentarily uncouple an affected point and neighbourhood, and the mathematical vector will be offset accordingly. If the invariant manifold of the whole system is a 3-torus, it was shown, that the trajectory of this mathematical vector can change almost randomly under these perturbations. Since these random perturbations act on particular points or areas in space and the trajectory of the system describes its development in time, this means nothing else than uncoupling in space leads to uncoupling in time.

Clearly, the argument for a 3-torus as invariant manifold of a system to cause chaos or turbulence can be derived from the fact that every "physical" space is three-dimensional (i.e., 3D world). In an n-dimensional physical space, the projection of an n-torus would be necessary to create the type of structural instability against random perturbations that would lead to chaotic behaviour. At the present level of the study, virtually nothing specific can be said about the requirements on random perturbation combinations, which can cause a drastic change in the state space trajectory. However, it is evident that not any combination of random perturbation can effect such a change in trajectory. This seems to relate in some way to the requirements Ruelle and Takens impose on the parameters p_1, p_2, q_1, q_2, which are used in the definition of their vector field z (see page 17) and the fact that the topology specifying the type of neighbourhood between the vector fields z and ω cannot be any "reasonable" topology but must be C^2.

Another interesting question in connection with appropriate random perturbation combinations would be how the probability for the occurrence of such a combination changes with the level of discretization (coarser of finer) and the type of probability distributions used on the random number generators.

This question also touches the problem of reality of the numerical modelling or simulation of the basic theoretical model. As in most studies involving fluid dynamic numerical simulation, substantial compromises, simplifications and approximations are forced by the complexity of the problem at hand (see e.g., the studies of the cavity flow problem mentioned in Appendix D and discretization of NS-equations in Chapter 5).

One of the biggest simplifications is the discretization itself. Mathematically, an infinite-dimensional function space is approximated by a finite-dimensional one, that is, by a set of discrete functions. The main problem here is a drastic change of compactness properties, and it has to be shown for a particular type of discretization, that the finite-dimensional approximated spaces have the right upper limit, i.e., the correct infinite-dimensional solution space. A considerable amount of this theoretical groundwork is collected in Temam [12]. On the other hand, it can be argued on physical grounds that such an approximation of an infinite-dimensional space must be reasonable, since in a fluid domain, each point with its associated variables such as pressure, temperature, velocity, etc. can be considered a quite accurate representative for a certain neighbourhood around it, as long as this neighbourhood is chosen small enough. Therefore, the selection of a finite number of points, whose neighbourhoods define a partition on the fluid domain directly relates to the transition from an infinite-dimensional solution space to a more or less accurate finite-dimensional approximation, depending on the number of selected points (i.e. coarseness or level of discretization) and, therefore, on the size of their neighbourhoods. The use of finite-element techniques with appropriate overlapping basis functions or weight-functions can therefore be considered as just a technical trick for reducing the number of selected points further, without losing too much accuracy while incorporating certain conditions, such as incompressibility.

A further concession for the sake of technical feasibility has to be made by substituting a variational formulation for the original formulation of the problem, as a consequence of the employed Galerkin method. The variational formulation might permit more solutions than the original one.

The restriction to rather specific, i.e., Dirichlet boundary conditions on a bounded subdomain within the entire fluid domain for the time dependent perturbations is a direct consequence of the rather high generality for the types of mean-flow profiles. These profiles can be selected without any considerations for their underlying boundary conditions. The motivation for this approach was the fact that, in experiment, turbulent "spots" can be observed in an otherwise laminar environment, although in the present model, these "spots" have to be exactly rectangular due to the type of space discretization. This drawback, however, could be reduced by using finer discretization and identifying the rectangular boundaries for the time dependent perturbations with the boundaries of the mean flow under an appropriate flow profile.*

Another important factor, which requires further investigation, is, as already mentioned, the level of discretization and the shape and size of the underlying rectangular elements, which can be varied along the three axes of the domain under consideration. This would vary the level of discretization within the domain spatially, and thereby permit to focus the study onto the more critical and interesting areas of the flow. In connection with this local and global varying of the discretization level, it would be interesting to observe how the critical value of the bifurcation parameter and other output parameters of the bifurcation program BIFOR 2 change, especially the approximation for $\vec{V}_1$ and solution trajectory for $\vec{V}_2$.

---

*The simple geometric configuration for such a study would be the one of the already mentioned cavity flow (see Appendix D).

Also, with the local and global varying of the discretizations, an analysis of the bifurcational behaviour of different mean flow profiles would be important. Amongst the four profiles tested in this study the ones with the more interesting results, i.e., Flows III and IV, seem to be too complex in their profiles to be realistic. That is to say, simpler profiles might show a more interesting bifurcational behaviour for a higher level of discretization. The main deficiency of the numerical simulation, i.e. stability of mean flow $\vec{U}$ for negative viscosities only (apart from the case of exception mentioned in Appendix C) could be a consequence of the downgrading induced by the discretization as described previously, such as: Dirichlet boundary conditions, variational principle, low level of discretization. However, the main reason for this phenomena is most likely the neglect of a pressure perturbation gradient. That would mean absence of a driving force, which is compensated for by negative viscosities thereby creating a driving force.

However, apart from this negative viscosity and despite the fact that, for reasons of economy, the lowest possible level of discretization, i.e., 4 points, had to be chosen, the results show in general a qualitatively correct behaviour for flow IV. The trajectories of $\vec{V}_1 + \vec{V}_2$ clearly show an increase of the level of instability of $\vec{V}_1$ when increasing the value of the bifurcation parameter η. The physically relevant range for η seems to be $.242 \leq \eta < .440$, i.e., between the critical value of the first Hopf bifurcation and the value for occurrence of unbounded growth. It can be assumed that beyond $\eta \sim .449$ the shear of flow IV becomes unrealistically high or at least too high for the low level of discretization. Moreover, since

$\vec{V}_1$ is represented by its first-order approximation, which is exact only at $\eta = \eta_c = .242$, its inaccuracies at $\eta \sim .449$ might be too high and therefore contribute to a breakdown of the model.

The most interesting and most promising parameter-range, as far as compliance with the hypothesis of a 3-torus as invariant manifold is concerned, seems to be the neighbourhood of $\eta \sim .3$. The unperturbed trajectory projections give reason to suspect the presence of at least two frequencies, which are close to each other and non-rationally related. Whether this is only a transitional stage of the trajectories between the orbit for $\vec{V}_1$ and a double-loop limit cycle, as can be seen for trajectories at $\eta > .33$, would require monitoring over considerably longer time intervals as could be done with the CPU time available. The fact that the randomly perturbed trajectories even for $\eta > .33$ show no tendency to approach such a double-loop limit-cycle suggests that if such a limit-cycle is present for $\eta < .33$, it must be a very weak attractor if any attractor at all.

The perturbed spectra, on the other hand, give reason to assume the range of turbulence and, therefore, the existence of a 3-torus as invariant manifold, for a parameter η between $\sim .33$ and at least $\sim .445$. However, this has to be looked upon with a certain amount of suspicion and doubt, since, as already mentioned in the discussion of the spectra, the sample sizes had to be kept too small to permit definite conclusions about the spectra and their slopes. For a comparison to experimentally obtained spectra, the sample size would have to be increased to at least 1.2×10^4, which is about the lower limit for experimentally obtained samples. Wind measurements for spectral analysis are usually done over periods from 20 minutes up to one hour, with a number of 10 to as many as 100 measurements per second and subsample sizes covering one minute. This gives overall sample sizes from 1.2×10^4 up to 3.6×10^5. For smaller sample sizes smoothing difficulties are encountered, so a definite spectral slope cannot be determined. Due to restrictions on available CPU time, the numerical analysis in general and the spectral analysis in particular of this study could not be advanced beyond its present level, which, especially in comparison with the orders of magnitude of sample sizes for experimentally obtained spectra, cannot be considered anything more than preliminary.

REFERENCES

[1] Tritton, D.J.: Physical Fluid Dynamics. Van Nostrand Reinhold, 1977.

[2] D. Ruelle and F. Takens: On the Nature of Turbulence. Commun. Math. Phys. 20, 167-192 (1971).

[3] Landau, L.D. and Lifschitz, E.M.: Fluid mechanics. Oxford: Pergamon, 1959.

[4] Marsden, J.E. and McCracken, M.: The Hopf Bifurcation and its Applications. Springer Verlag, New York, 1976.

[5] Hassard, B.D., Kazarinoff, N.D. and Wan, Y.-H.: Theory and Application of Hopf Bifurcation. London Math. Society Lecture Notes Series, 1981.

[6] Hopf, E.: Abzweigung einer periodischen Lösung Von einer stationären Lösung eines Differential Systems. Ber. Math. -Phys. K.L. Sächs. Akad. W.SS. Leipzig 94, 1-22 (1942).

[7] Chenciner, A. and Iooss, G.: Bifurcation de tores invariantes. Arch. Rat. Mech., Vol. 69, #2, 109-198, 1979.

[8] Sell, G.R.: Bifurcation of Higher Dimensional Tori. Arch. Rat. Mech. 69, 199-230 (1979).

[9] Chillingworth, D.R.J.: Differential topology with a view to applications. Pitman, London, 1976.

[10] Smale, S.: Differentiable dynamical systems. Bull. Am. Math. Soc. 73, 747-817 (1967).

[11] Fortin, M.: Calcul numerique des éculements des fluides de Bingham et des fluides newtoniens incompressible par la méthode des éléments finis. Thèse, Université de Paris, 1972.

[12] Temann, R.: Navier-Stokes Equations. North-Holland Publishing Co., 1979.

[13] Kolmogorov, A.N.: The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. C.R. Acad. Sci. U.R.S.S., 30, 301 (1941).

[14] Heisenberg, W.: Zeitschrift fuer Physik 124, 614 (1948). Proc. Roy. Soc. (London) A195 402 (1948,49).

[15] Morris, D.J.: Solution of the incompressible driven cavity problem by the alternating direction implicit method. NASA-SP-378, N76-16028, (1976), p. 47-60.

[16] Smith, R.E., Kidd, A.: Comparative study of two-numerical techniques for solution of viscous flow in a driven cavity. NASA-SP_378, N76-16029, (1976), p. 61-82.

[17] Rubin, S.G., Graves, R.A.: Viscous flow solution with a cubic spline approximation. Computers and Fluids, (1975), Vol. 3, p. 15-31.

[18] Rubin, S.G., Graves, R.A.: Cubic spline solution for the driven cavity. NASA-SP-378, N76-16032, (1976), p. 119-132.

[19] Boney, L.R.: Application of upstream weighted-differencing to the driven cavity problem. NASA-SP-378, N76-16025, (1976), p. 7-22.

[20] Hefner, J.N.: Solution of the incompressible driven cavity problem cavity problem using the CROCCO method. NASA-SP-378, N76-1626, (1976), p. 23-32.

[21] Hirsh, R.S.: Solution of the driven cavity problem in primitive variables using the SMAC method. NASA-SP-378, N76-16027, (1976) p. 33-46.

[22] Suttles, J.T.: Calculations of steady viscous flow in a square driven cavity by the artificial compressibility method. NASA-SP-378, N76-16030, (1976), p. 83-102.

[23] Zoby, E.V.: Finite difference solution for the incompressible driven cavity flow problem. NASA-SP-378, N76-16031, (1976), p. 103-118.

[24] Cheng, R. Ta-Shun: Solution of the Navier-Stokes equations by the finite-element method. The Physics of Fluids, 1972, Vol. 15, No. 12, p. 2098-2105.

[25] Taylor, C., Hood, P.: A numerical solution of the Navier-Stokes equations using the finite-element technique. Computers and Fluids, 1973, Vol. 1, p. 73-100.

[26] Kawahara, M., Koshimura, N., Nakogawa, K., Oheaka, H.: Steady and unsteady finite element analysis of incompressible viscous fluid. Int. Numer. Math. in Engrg., 1976, Vol. 10, p. 437-448.

[27] Fortin, M., Thomasset, F.: Mixed finite-element methods for incompressible flow problems. Journal of Computational Physics, 1979, Vol. 31, p. 113-145.

[28] Olson, M.D., Tuann, S.-Yu: New finite-element results for the square cavity. Computers and Fluids, 1979, Vol. 7, p. 123-135.

[29] Shampine, L.F., Watts, H.A., Davenport, S.M.: Solving nonstiff ordinary differential equations - the state of the art. SIAM Review: Vol. 18, No. 3, 1976.

APPENDIX A

Some of the concepts used in Chapter 2 and 3 are explained here. Except for the description of the Cantor set and the strange attractor, all definitions and some of their interpretations are adopted from [9]:

1. THE CANTOR SET

Construction

The best known way of constructing a Cantor set is to trisect the closed unit interval $I = [0,1]$ at the points $1/3$ and $2/3$ and then delete the open interval $(1/3, 2/3)$, called the 'middle third'. C_1 denotes here the remainder of the points in I, i.e.,

$$C_1 = [0,1/3] \cup [2/3, 1]$$

We now trisect each of the two segments in C_1 at $1/9$ and $2/9$ and $7/9$ and $8/9$, and then delete the 'middle third' from each segment, i.e, $(1/9, 2/9)$ and $(7/9, 8/9)$. Let C_2 denote the remainder of the points in C_1, i.e.,

$$C_2 = [0,1/9] \cup [2/9,1/3] \cup [2/3,7/9] \cup [8/9,1]$$

If we continue in this manner we obtain a descending sequence of sets

$$C_1 \supset C_2 \supset C_3 \supset \ldots$$

where C_m consists of the points in C_{m-1} excluding the 'middle thirds'.

Observe that C_m consists of 2^m disjoint closed intervals and, if we number them consecutively from left to right, we can speak of the odd or even intervals in C_m. The Cantor set C is the intersection of these sets, i.e.

$$C = \cap\{C_i : i \in N\}$$

Some Properties

1. C is non denumerable. To see this, define a function f on C as follows:

$$f(x) = \langle a_1, a_2, \ldots \rangle$$

where

$$a_m = \begin{cases} 0 & \text{if } x \text{ belongs to an odd interval in } C_m \\ 2 & \text{if } x \text{ belongs to an even interval in } C_m \end{cases}$$

C is equivalent to the set of sequences $\langle a_1, a_2, \ldots \rangle$, where $a_i = 0$ or 2, which has cardinality $2^{\text{aleph null}}$ which is the continuum. Therefore every point in C is an accumulation point.

2. C has measure zero. The measure of the complement of C relative to $I = [0,1]$, i.e., the union of the middle thirds, equals

$$\frac{1}{3} + \frac{2}{9} + \frac{4}{27} + \frac{8}{81} + \ldots = 1$$

Since the measure of I is also 1, the measure of C must be zero.

2. THE HORSESHOE DIFFEOMORPHISM*

The present introduction is rather informal. For its original introduction, see the paper by Smale [10], who discovered this diffeomorphism in 1966.

Take a rectangle R in the plane $\mathbb{R}^2$. Stretch it and bend it into a horseshoe shape as shown in Figure 5 (vertices A,B,C,D go to A',B',C',D') and place it on top of its original position. This can be done mathematically using smooth but possibly nonanalytic functions pieced together to obtain the proper diffeomorphism $f : \mathbf{R}^2 \to \mathbf{R}^2$. It is important that this diffeomorphism stretches the two rectangular strips $R \cap f^{-1}(R)$ (consisting of points of R which remain in R after f is applied) linearly by a factor > 1 in the vertical direction (parallel to AD) and compresses these strips by a factor < 1 in the horizontal direction (parallel to AB).

Now look at $R \cap f^{-1}(R) \cap f^{-2}(R)$, i.e., the set of points which remain in R when both f and f^2 are applied to them. This consists of four thin rectangles, two in each rectangle of $R \cap f^{-1}(R)$. Similarly, $R_{-n} = R \cap f^{-1}(R) \cap \ldots \cap f^{-n}(R)$ consists of 2^n thin rectangles contained pairwise in the 2^{n-1} rectangles of $R_{-(n-1)}$ (see Figure 6).

The set

$$R_{-\infty} = R \cap f^{-1}(R) \cap f^{-2}(R) \cap \ldots,$$

which consists of those points of R which remain in R after any number of positive iterations of f, is the cartesian product $\overline{AB} \times C$,

---

*A diffeomorphism is a differentiable mapping with differentiable inverse.

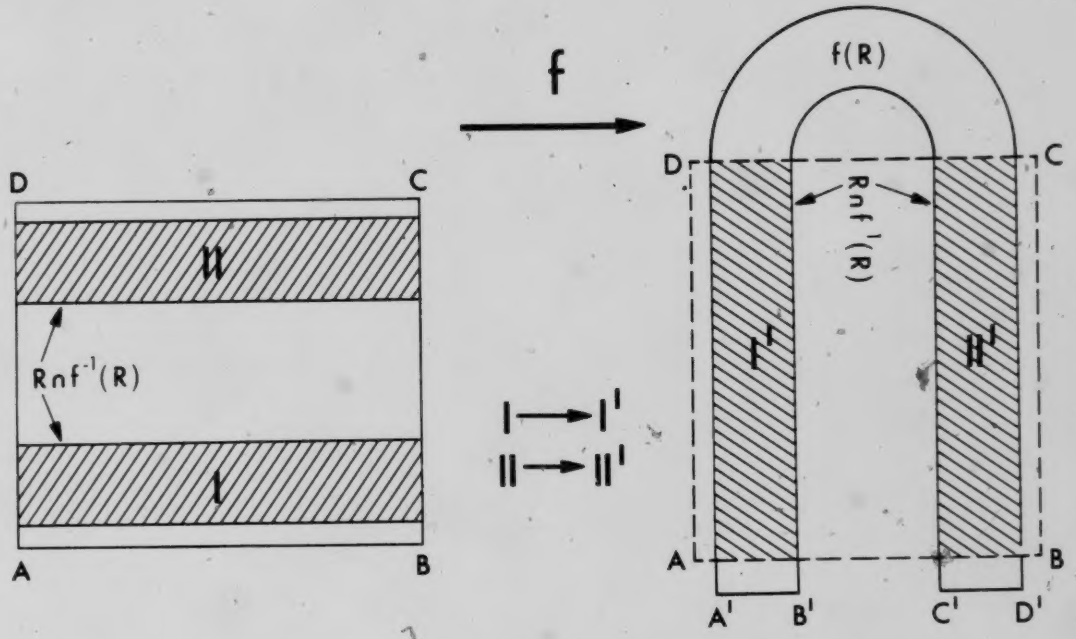

f
f(R)
D
C
II
Rnf⁻¹(R)
I
A
B
I → I'
II → II'
D
C
Rnf⁻¹(R)
I'
II'
A
B
A'
B'
C'
D'

Figure 5. Horseshoe-Diffeomorphism.
First Mapping

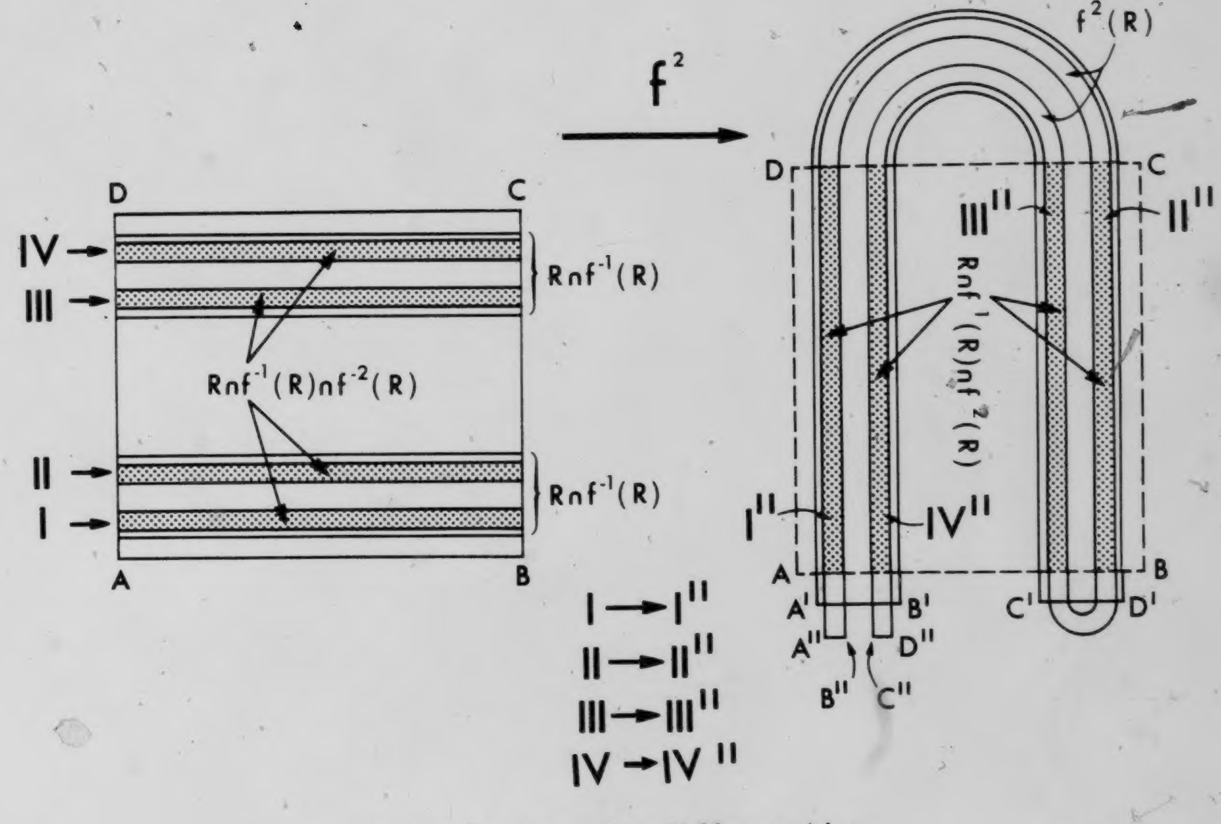

f²
f²(R)
D
C
IV
III
Rnf⁻¹(R)
Rnf⁻¹(R)nf⁻²(R)
II
I
Rnf⁻¹(R)
A
B
I → I''
II → II''
III → III''
IV → IV''
D
C
III''
II''
Rnf⁻¹(R)nf⁻²(R)
I''
IV''
A
B
A'
B'
C'
D'
A''
D''
B''
C''

Figure 6. Horseshoe-Diffeomorphism.
Second Mapping

where $\overline{AB}$ is the closed interval corresponding to the width AB of R and C is an example of a Cantor set. Clearly the diffeomorphism f could be constructed to have this Cantor set develop exactly like the one constructed by taking out the middle thirds. Likewise, the set R_∞ consisting of those points of R which remain in R under all negative iterations of f is of the form $C \times \overline{AB}$. If Λ denotes $R_{-\infty} \cap R_\infty$ then we see that Λ is a closed invariant set for f, meaning that f takes Λ onto itself, and Λ is homeomorphic to $C \times C$, which can be shown to be in fact homeomorphic to C. Furthermore, it can be shown that:

(1) There is an infinite number of periodic points of f in Λ.

(2) There are points which have orbits that are dense in Λ.

(3) The effect of f near Λ is to contract in a horizontal direction and expand in a vertical direction.

Clearly, this construction can be modified using a disk instead of rectangles and a diffeomorphism f such that $f(R) \subset R$ (see Figure 7). This is the variation used by Ruelle and Takens and it is easy to find a subset homeomorphic to the rectangle ABCD, which develops in the way described above (see Figure 6).

3. NONWANDERING SETS

Definition

A point $x \in X$ is said to be *nonwandering* with respect to a mapping f if for every neighbourhood U of x, there exists an $n > 0$ with $U \cap f^{\pm n}(U) \neq \emptyset$. The set of all nonwandering points is called the *nonwandering set*. That means, the point x under the mapping f will sooner or later return arbitrarily close to any position it ever takes or took under f. E.g., for a quasiperiodic motion on the k-torus T^k we have T^k itself as the nonwandering set.

4. STRUCTURAL STABILITY

In the following definitions we use the notations:

$X^r(M)$ is the set of all vector fields differentiable up to at least r^{th} order defined on the manifold M.

$\mathrm{Diff}^k(M)$ is the set of all diffeomorphisms differentiable up to at least k^{th} order defined on the manifold M.

Definition

Two vector fields $x, y \in X^r(M)$ are C^k *equivalent* if there exists $h \in \mathrm{Diff}^k(M)$ mapping x-trajectories to y-trajectories (and vice versa), preserving their sense of motion, but not necessarily their parameterization by time.

Definition

A vector field $x \in X^r(M)$ is *structurally stable* if there is a neighbourhood N of x in $X^r(M)$ $(N \subset X^r(M))$ such that every vector field in N is C^o equivalent (i.e., topologically equivalent or homeomorphic) to x. This means that the (topological) structure of the trajectories of x does not change if x is slightly perturbed so that it becomes some other vector field x', say, which is still an element of the neighbourhood N. In the most simple terms: if any vector field close enough to x has trajectories very much like those of x, then the vector field x is structurally stable.

5. MORSE-SMALE SYSTEMS

Definition

A fixed point p of a vector field x is called *hyperbolic*, if the linearized vector field at p has no purely imaginary or zero eigenvalue. That means the eigenvalues of the linearized field x at p must have either positive or negative real parts, respectively. Therefore, the vector field x in a certain neighbourhood U of p must have components either radially directed outwards or inwards, respectively, w.r.t. p.

The set of all trajectories of a vector field is sometimes called the *flow* and denoted by ϕ. More specifically, $\phi_t(p)$ is used to denote the point up to which the point p "travels" within the time t under the influence of the vector field. Looking at the obvious properties

$$\phi_t \cdot \phi_{-t} = \phi_o = \text{identity}$$

$$(\phi_t \cdot \phi_s) \cdot \phi_u = \phi_t \cdot (\phi_s \cdot \phi_u) = \phi_{t+s+u} \, ,$$

it is evident that ϕ is a commutative Lie group.

Definition

Let ϕ be the flow of a vector field x on a manifold M and p a hyperbolic fixed point. The set

$$W \pm (p) = \{q \in M \mid \phi_t(q) \to p \text{ for } t \to \pm\infty\}$$

is called the *stable* (for +) or *unstable* (for -) *manifold* for p.

Clearly, a point on the stable manifold for p is attracted towards p through the vector field x for positive times, whereas a point on the unstable one is attracted to p for negative times.

Definition

A vector field $x \in X^r(M)$, is *Morse-Smale* (MS) if:

(1) The number of fixed points and periodic orbits is finite, and each is hyperbolic.

(2) The set of nonwandering points consists of fixed points and periodic orbits only.

(3) All stable and unstable manifolds intersect transversally.

For two-dimensional manifolds the following statements hold:

1. structural stability implies MS.
2. MS systems are dense in $\text{Diff}^1(M)$ or $X^1(M)$.
3. Structurally stable systems are dense in $\text{Diff}^1(M)$ or $X^1(M)$.

For manifolds with dimensions higher than 2 only MS $\Longrightarrow$ structural stability has been proved by Palis and Smale. The rest of the statements is false in general.

6. GENERICITY

Definition

Let S be any topological space. A subset G of S is called *residual* if it is the intersection of a countable number of sets, each of which is both open and dense in S.

The motivation behind this definition is to distinguish between two different types of denseness. For example, both the set of rational numbers $\mathbb{Q}$ and the set of irrational numbers $\mathbb{I}$ are dense in the set of real numbers $\mathbb{R}$. However, $\mathbb{I}$ is in the same sense "denser" in $\mathbb{R}$ then $\mathbb{Q}$; there are more irrationals than rationals. Since the rationals are countable:

$$\mathbb{Q} = \{q_1,q_2,\ldots,q_n,\ldots\} ,$$

one can express the irrationals as the countable intersection:

$$\mathbb{I} = \bigcap_{n=1}^{\infty} (\mathbb{R} - \{q_n\}).$$

This construction of the irrationals follows the definition; they are therefore a residual subset of the reals, whereas the rationals are not.

The definition of genericity is just a slight generalization of the previous one:

Definition

A property p of elements e of a set S* is called *generic* if these elements form a set P which contains a residual subset of S.

Replacing a real number by another real by adding an arbitrary "perturbation", the probability of the new real to be rational is zero whereas the probability to be irrational is one. That is, the generic property is adopted. This shows how the concept of genericity can be understood and applied in connection with the vector fields discussed at the end of Chapter 2.

---

*In general, S will contain elements other than the e as well.

APPENDIX B

1. THE DIRAC Δ-FUNCTION

The Δ-function can be defined as any distribution with the following properties:

(i) $\Delta_r(x) = 0 \quad \text{for} \quad x \neq r$

(ii) $\int \Delta_r(x)dx = 1$

(iii) $\int \Delta_r(x)f(x)dx = f(r)$.

The integration can go over more than one dimension and the range of integration naturally has to include the point r.

Analytic representations of $\Delta_r(x)$ are e.g.:

$$\Delta_r(x) = (2\pi)^{-1} \int_{-\infty}^{\infty} \exp[i\omega(x-r)]d\omega$$

or:

$$\Delta_r(x) = \lim_{\Omega \to \infty} \frac{\sin(\Omega x)}{\pi x}$$

In the present context, the property (iii) is used with the following notation:

$$\langle \Delta_r(x) \mid f(x) \rangle = f(r),$$

i.e., it is written as a scalar product of two vectors defined in function space.

2. CONSTRUCTION OF A LINEAR 10 × 10 SYSTEM FOR THE DETERMINATION OF THE COEFFICIENTS IN THE BASIS FUNCTIONS

As noted in the text, the basis functions $W_r(x)$, $r \in S$ are polynomials:

$$W_r(X) = C + \sum_{i}^{3} C_i X_i + \sum_{i \leq j}^{3} C_{ij} X_i X_j \qquad \text{(B1)}$$

$$X = (X_1, X_2, X_3)$$

With the following conditions:

Four Nodal Value Conditions

$$\begin{aligned} W_r(r) &= 1 \\ W_r(r_L) &= 0, \quad L = 1,2,3 \end{aligned} \qquad \text{(B2)}$$

Here the r_L denote the three centroids on the other three faces of the tetrahedron.

Three Conditions for Zero Divergence Requirement

$$\int_T \frac{\partial W_r(X)}{\partial X_i} dX_1 dX_2 dX_3 = 0, \quad i = 1,2,3 \qquad \text{(B3)}$$

(integration over one tetrahedron).

Three Gradient Conditions

$$\left.\frac{\partial W_r(X)}{\partial X_i}\right|_{X=r} = 0, \quad i = 1,2,3 \qquad \text{(B4)}$$

To apply these conditions, the basis functions W(X)* have to be transformed into barycentric coordinates.

Notations:

X with indices i,j,k denotes cartesian coordinates

with indices ℓ,m,n,o denotes barycentric coordinates

X with double indices i,j,k; ℓ,m,n,o denotes cartesian coordinates (i,j,k; first index) of the four tetrahedral vertices (ℓ,m,n,o; second index)

For the relations between cartesian and barycentric coordinates, we write

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} = \begin{bmatrix} x_{11} & x_{21} & x_{31} & x_{41} \\ x_{12} & x_{22} & x_{32} & x_{42} \\ x_{13} & x_{23} & x_{33} & x_{43} \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} \qquad \text{(B5)}$$

or:

$$X_i = \sum_{L}^{4} X_{iL}\lambda_L$$

$$1 = \sum_{L}^{4} \lambda_L \qquad \text{(B6)}$$

Using (B6), the polynomial (B1) and its derivatives

$$\frac{\partial W}{\partial X_k}(X) = C_k + \sum_{1}^{3} C_{ik}X_i + C_{kk}X_k, \quad k = 1,2,3 \qquad \text{(B7)}$$

---

[*] Subscript r dropped for conciseness.

can now be transformed directly:

$$W[X(\lambda)] = C + \sum_{1}^{3} C_i\left[\sum_{L}^{4} X_{iL}\lambda_L\right] + \sum_{i\leq j} C_{ij}\left[\sum_{L}^{4} X_{iL}\lambda_L\right]\left[\sum_{1}^{4} X_{ij}\lambda_L\right] \quad \text{(B8)}$$

$$\frac{\partial W}{\partial X_k}[X(\lambda)] = C_k + \sum_{i}^{3} C_{ik}\left[\sum_{L}^{4} X_{iL}\lambda_L\right] + C_{kk}\left[\sum_{L}^{4} X_{kL}\lambda_L\right] \quad \text{(B9)}$$

Relations (B5) or (B6), respectively show linear dependence between the barycentric coordinates λ_L. Therefore, in order to perform the integrations for the zero-divergence condition, one of the λ_L must be eliminated. The vertex associated with the eliminated barycentric coordinate is the origin of a skew coordinate system spanned by the three edges of this vertex.

Taking the vertex X_{i4}, $i = 1,2,3$ as origin, the integral of $\frac{\partial W}{\partial X_k}$ over the tetrahedron may be written as

$$I = \int_T \frac{\partial W}{\partial X_k}[X(\lambda_1,\lambda_2,\lambda_3)]\left[\frac{\partial X_i}{\partial \lambda_L}\right] d\lambda_1 d\lambda_2 d\lambda_3 \quad \text{(B10)}$$

Here $\left[\frac{\partial X_i}{\partial \lambda_L}\right]$ is the functional determinant with $L \neq 4$.

Eliminating λ_4 from (B6) gives:

$$\begin{aligned} X_i &= \sum_{L}^{3} X_{iL}\lambda_L + X_{i4}(1 - \lambda_1 - \lambda_2 - \lambda_3) \\ &= \sum_{L}^{3} (X_{iL} - X_{i4})\lambda_L + X_{i4} \end{aligned} \quad \text{(B11)}$$

therefore:

$$\left[\frac{\partial X_i}{\partial \lambda_L}\right]_{L\neq 4} = \begin{vmatrix} X_{11} - X_{14} & X_{21} - X_{24} & X_{31} - X_{34} \\ X_{12} - X_{14} & X_{22} - X_{24} & X_{32} - X_{34} \\ X_{13} - X_{14} & X_{23} - X_{24} & X_{33} - X_{34} \end{vmatrix}$$

$$\equiv [X_{iL} - X_{i4}]_{\det} \qquad \text{(B12)}$$

(B9), (B11), (B12) into (B10):

$$I = [X_{iL}-X_{i4}]_{\det} \int_T \left[C_k + \sum_i^3 C_{ik}\left[\sum_L^3 (X_{iL}-X_{i4})\lambda_L + X_{i4}\right]\right.$$

$$\left. + C_{kk}\left[\sum_L^3 (X_{iL}-X_{i4})\lambda_L - X_{i4}\right]\right] d\lambda_1 d\lambda_2 d\lambda_3$$

$$= [X_{iL}-X_{i4}]_{\det}\left[I_o C_k + \sum_i^3 C_{ik}\left[\sum_L^3 (X_{iL}-X_{i4}) I_L + I_o X_{i4}\right]\right.$$

$$\left. + C_{kk}\left[\sum_L^3 (X_{iL}-X_{i4}) I_L + I_o X_{i4}\right]\right] \qquad \text{(B13)}$$

with:

$$I_o = \int_T d\lambda_1 d\lambda_2 d\lambda_3 = \frac{1}{6}$$

$$I_L = \int_T \lambda_L d\lambda_1 d\lambda_2 d\lambda_3 = \frac{1}{24} \qquad \text{(B14)}$$

The integrals I_o, I_L are most easily evaluated by use of Hammer's formula:

$$\int_T \lambda_1^{p_1} \lambda_2^{p_2} \lambda_3^{p_3} d\lambda_1 d\lambda_2 d\lambda_2 = \frac{p_1! p_2! p_3!}{(p_1+p_2+p_3+3)!} \qquad \text{(B15)}$$

The equations for the ten conditions (B2), (B3), (B4) can now be written down explicitly:

4 Nodal Value Conditions:

$W(r_4) = 1$:

r_4 is the centroid on the face opposite to the barycentric origin (X_{14}, X_{24}, X_{34}). Using (B8) and r_4 $(\lambda_1, \lambda_2, \lambda_3) = (1/3, 1/3, 1/3)$; $\lambda_4 = 0$:

$$1 = C + \sum_{i}^{3} C_i \left[\frac{1}{3} \sum_{L}^{3} X_{iL}\right] + \sum_{i \le j}^{3} C_{ij} \left[\frac{1}{3} \sum_{L}^{3} X_{iL}\right]\left[\frac{1}{3} \sum_{L}^{3} X_{jL}\right] \qquad \text{(B16)}$$

$W(r_3) = 0$:

Coordinates of r_3 : $\lambda_1 = \lambda_2 = \lambda_4 = \frac{1}{3}$, $\lambda_3 = 0$

$$0 = C + \sum_{i}^{3} C_i \left[\frac{1}{3} \sum_{L=1,2,4} X_{iL}\right] + \sum_{i \le j}^{3} C_{ij} \left[\frac{1}{3} \sum_{L=1,2,4} X_{iL}\right]\left[\frac{1}{3} \sum_{L=1,2,4} X_{jL}\right] \qquad \text{(B17)}$$

$W(r_2) = 0$:

Coordinates of r_2 : $\lambda_1 = \lambda_3 = \lambda_4 = \frac{1}{3}$, $\lambda_1 = 0$

$$0 = C + \sum_{i}^{3} C_i \left[\frac{1}{3} \sum_{L=1,3,4} X_{iL}\right] + \sum_{i \le j}^{3} C_{ij} \left[\frac{1}{3} \sum_{L=1,3,4} X_{iL}\right]\left[\frac{1}{3} \sum_{L=1,3,4} X_{jL}\right] \qquad \text{(B18)}$$

$W(r_1) = 0$:

Coordinates of r_1 : $\lambda_2 = \lambda_3 = \lambda_4 = \frac{1}{3}$, $\lambda_1 = 0$

$$0 = C + \sum_{i}^{3} C_i\left[\frac{1}{3}\sum_{L=2,3,4} X_{iL}\right] + \sum_{i\le j}^{3} C_{ij}\left[\frac{1}{3}\sum_{L=2,3,4} X_{iL}\right]\left[\frac{1}{3}\sum_{L=2,3,4} X_{jL}\right] \tag{B19}$$

3 Zero Gradient Conditions:

Using Equation (B9) $(k = 1,2,3)$ and $\dfrac{\partial W(r_4)}{\partial X_k} = 0$

$$0 = \frac{1}{6} C_k + \sum_{i}^{3} C_{ik}\left[\frac{1}{3}\sum_{L}^{3} X_{iL}\right] + C_{kk}\left[\frac{1}{3}\sum_{L}^{3} X_{iL}\right], \qquad k = 1,2,3 \tag{B20}$$

3 Zero Divergence Conditions over Tetrahedron:

Using Equation (B13) $(k = 1,2,3)$ and $I = 0$:

$$0 = \frac{1}{6} C_k + \sum_{i}^{3} C_{ik}\left(\frac{1}{24}\sum_{L}^{4} X_{iL}\right) + C_{kk}\left(\frac{1}{24}\sum_{L}^{4} X_{iL}\right) \qquad k = 1,2,3 \tag{B21}$$

introducing the technical substitution:

$$S_i \equiv \sum_{L}^{4} X_{iL}$$

the linear system for the coefficients can be written down.

$$\left.\begin{aligned}
&1)\quad 0 = C + \frac{1}{3}\sum_{i}^{3} C_i[S_i - X_{i1}] + \frac{1}{9}\sum_{i\le j}^{3} C_{ij}[S_i - X_{i1}][S_j - X_{j1}]\\
&2)\quad 0 = C + \frac{1}{3}\sum_{i}^{3} C_i[S_i - X_{i2}] + \frac{1}{9}\sum_{i\le j}^{3} C_{ij}[S_i - X_{i2}][S_j - X_{j2}]\\
&3)\quad 0 = C + \frac{1}{3}\sum_{i}^{3} C_i[S_i - X_{i3}] + \frac{1}{9}\sum_{i\le j}^{3} C_{ij}[S_i - X_{i3}][S_j - X_{j3}]\\
&4)\quad 1 = C + \frac{1}{3}\sum_{i}^{3} C_i[S_i - X_{i4}] + \frac{1}{9}\sum_{i\le j}^{3} C_{ij}[S_i - X_{i4}][S_j - X_{j4}]
\end{aligned}\right\}\ \text{Nodal Values Conditions}$$

$$\left.\begin{aligned}
&5)\quad 0 = C_1 + \tfrac{2}{3}\,C_{11}[S_1-X_{14}] + \tfrac{1}{3}\,C_{12}[S_2-X_{24}] + \tfrac{1}{3}\,C_{31}[S_3-X_{34}] \\
&6)\quad 0 = C_2 + \tfrac{2}{3}\,C_{22}[S_2-X_{24}] + \tfrac{1}{3}\,C_{12}[S_1-X_{14}] + \tfrac{1}{3}\,C_{23}[S_3-X_{34}] \\
&7)\quad 0 = C_3 + \tfrac{2}{3}\,C_{33}[S_3-X_{34}] + \tfrac{1}{3}\,C_{23}[S_2-X_{24}] + \tfrac{1}{3}\,C_{31}[S_1-X_{14}]
\end{aligned}\right\}\ \text{Zero Gradient Condition}$$

$$\left.\begin{aligned}
&8)\quad 0 = \tfrac{1}{6}\,C_1 + \tfrac{1}{12}\,C_{11}S_1 + \tfrac{1}{24}\,C_{12}S_2 + \tfrac{1}{24}\,C_{31}S_3 \\
&9)\quad 0 = \tfrac{1}{6}\,C_2 + \tfrac{1}{12}\,C_{22}S_2 + \tfrac{1}{24}\,C_{12}S_1 + \tfrac{1}{24}\,C_{23}S_3 \\
&10)\quad 0 = \tfrac{1}{6}\,C_3 + \tfrac{1}{12}\,C_{33}S_3 + \tfrac{1}{24}\,C_{23}S_2 + \tfrac{1}{24}\,C_{31}S_1
\end{aligned}\right\}\ \text{Zero Divergence Condition} \qquad \text{(B22)}$$

The system is written down for the basis function $W_{r_4}(X)$, i.e., the one which takes value 1 at the centroid in the face opposite to vertex No. 4. It can be converted to the systems for $W_{r_1}(X)$, $W_{r_2}(X)$, $W_{r_3}(X)$ by replacing the seond index (i.e., 4) on X in the zero gradient conditions by 1,2,3 respectively and by permuting the 1 on the left hand side of Equation (4) in the system to the left hand side of Equation 1), 2), 3), respectively.

3. THE PROGRAM DSCRB

DSCRB discretizes a given three-dimensional rectangular domain D into tetrahedrons, calculates the ten coefficients for the basis functions of all nodes according to the linear system derived in Section 2 and uses them in the basis functions to calculate the matrix elements for the interior nodes* in Equation (5.14) for a given flow $\vec{U}$ of the structure (7.3). Finally, it inverts the matrix on the left hand side over to the right hand side to obtain Equation (5.20).

---

*To induce Dirichlet boundary conditions, the nodes on ∂D, consisting of the surface planes of D, are omitted.

The program ADV complements DSCRB by reassembling the nonlinear matrix elements W^{∇}_{onm} calculated by DSCRB into the additional advective matrix $W^{\nabla}_{onm} + W^{\nabla}_{omn}$ in the Equations (5.18), (5.19) for the second and third perturbations $\vec{V}_2$ and $\vec{V}_3$ and multiplying it with the inverse of the left hand side matrix W_{on}.

3.1 Discretization of the Domain D in the Main Line Program*

The program begins by partitioning a rectangular domain D into blocks. The size of these blocks and D is naturally induced by three finite sets of user-supplied discrete values along the three coordinate axes (XIN, YIN, ZIN). At the same time, each one of these blocks is split into five tetrahedrons according to configuration A or B (see Figures 12a,b). These two configurations are alternating between two blocks sharing a face in order to obtain an admissible discretization (see Temam [12], p. 73), which means that directly neighbouring tetrahedrons have to have common faces and edges. The splitting of the individual blocks is simply done by reassigning the block corners as corners to the five tetrahedrons, which are numbered at the same time from one to five. Next, for each individual tetrahedron the nodes (i.e., centroids of the faces) are identified by their opposite vertices, the number of the tetrahedron and the coordinates of the block (identified by the three numbers of discrete values along the three axes, which determine the left bottom-front corner of the block). To determine row and column numbers for the linear matrices, the nodes are being counted through as they are being identified. At this stage each node is checked using its identification whether it has already been counted in a previously encountered tetrahedron or whether it is located in ∂D. If this is the case, the node is skipped by the count. This is possible since every node is shared by two tetrahedrons (except nodes in ∂D) and carries therefore two identifications. Once the program established one identification, it can be used to determine the other one and thereby whether it was previously encountered. After this counting and checking procedure, the linear matrix elements are calculated by subroutines and indexed according to the count numbers of the nodes and their place in the fluid space subdecompositions.

---

*See also 3.10 for details.

3.2 Matrix Elements for Advective Terms

Before the matrix generation of the program can be described, it is necessary to analyse the particular structure of the advective matrix elements $W_{on}^{U\nabla}$ and $W_{on}^{\nabla:U}$ (see (5.14) and (5.15)) for a mean flow $\vec{U}$ of the type (7.3).

First, $W_{on}^{U\nabla}$ and $W_{on}^{\nabla:U}$ can be combined into one matrix element:

$$W_{on}^{U\nabla} + W_{on}^{\nabla:U} = < W_{r_o}(X) I_3 | A_d > \qquad \text{(B23)}$$

with

$$A_d \equiv (\vec{U}[\vec{\nabla} : W_{r_n}(X)] + [\vec{\nabla} : \vec{U}] W_{r_n}(X)) \qquad \text{(B24)}$$

as the discretized version of the advective operator defined in (4.3). Written out in components by use of (5.10a), (5.10b) A_d looks like:*

---

*$W_{r_n}(X)$ is simply written as W for reasons of conciseness.

$$A_d = \begin{bmatrix} W\frac{\partial U_1}{\partial X_1} + [\vec{U}\cdot\vec{\nabla}W] & W\frac{\partial U_1}{\partial X_2} & W\frac{\partial U_1}{\partial X_3} \\ W\frac{\partial U_2}{\partial X_1} & W\frac{\partial U_2}{\partial X_2} + [\vec{U}\cdot\vec{\nabla}W] & W\frac{\partial U_2}{\partial X_3} \\ W\frac{\partial U_3}{\partial X_1} & W\frac{\partial U_3}{\partial X_2} & W\frac{\partial U_3}{\partial X_3} + [\vec{U}\cdot\vec{\nabla}W] \end{bmatrix} \tag{B.25}$$

with:

$$\vec{U}\cdot\vec{\nabla}W = U_1\frac{\partial W}{\partial X_1} + U_2\frac{\partial W}{\partial X_2} + U_3\frac{\partial W}{\partial X_3} \ . \tag{B.26}$$

Substituting the structure of (7.3) for $\vec{U}$ gives:

$$\begin{aligned} A_d = {} & C_1M_1 + C_2M_2 + C_3M_3 + C_{11}M_{11} + C_{22}M_{22} + C_{33}M_{33} \\ & + C_{12}M_{12} + C_{21}M_{21} + C_{23}M_{23} + C_{32}M_{32} + C_{12}M_{13} + C_{31}M_{31} \end{aligned} \tag{B.27}$$

with:

$$M_1 = \frac{\partial W}{\partial X_1}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad M_2 = \frac{\partial W}{\partial X_2}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad M_3 = \frac{\partial W}{\partial X_3}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_{11} = \begin{bmatrix} WE_{11}X_1^{E_{11}-1} + X_1^{E_{11}}\frac{\partial W}{\partial X_1} & 0 & 0 \\ 0 & X_1^{E_{11}}\frac{\partial W}{\partial X_1} & 0 \\ 0 & 0 & X_1^{E_{11}}\frac{\partial W}{\partial X_1} \end{bmatrix}$$

$$M_{22} = \begin{bmatrix} X_2^{E_{22}} \frac{\partial W}{\partial X_2} & 0 & 0 \\ 0 & WE_{22}X_2^{E_{22}-1} + X_2^{E_{22}} \frac{\partial W}{\partial X_2} & 0 \\ 0 & 0 & X_2^{E_{22}} \frac{\partial W}{\partial X_2} \end{bmatrix}$$

$$M_{33} = \begin{bmatrix} X_3^{E_{33}} \frac{\partial W}{\partial X_3} & 0 & 0 \\ 0 & X_3^{E_{33}} \frac{\partial W}{\partial X_3} & 0 \\ 0 & 0 & WE_{33}X_3^{E_{33}-1} + X_3^{E_{33}} \frac{\partial W}{\partial X_3} \end{bmatrix}$$

$$M_{12} = \begin{bmatrix} X_2^{E_{12}} \frac{\partial W}{\partial X_1} & E_{12}X_2^{E_{12}-1} \cdot W & 0 \\ 0 & X^{E_{12}} \frac{\partial W}{\partial X_1} & 0 \\ 0 & 0 & X_2^{E_{12}} \frac{\partial W}{\partial X_1} \end{bmatrix}$$

$$M_{21} = \begin{bmatrix} X_1^{E_{21}} \frac{\partial W}{\partial X_2} & 0 & 0 \\ E_{21}X_1^{E_{21}-1} \cdot W & X_1^{E_{21}} \frac{\partial W}{\partial X_2} & 0 \\ 0 & 0 & X_1^{E_{21}} \frac{\partial W}{\partial X_2} \end{bmatrix}$$

$$M_{23} = \begin{bmatrix} X_3^{E_{23}} \dfrac{\partial W}{\partial X_2} & 0 & 0 \\ 0 & X^{E_{23}} \dfrac{\partial W}{\partial X_2} & E_{23} X_3^{E_{23}-1} \cdot W \\ 0 & 0 & X_3^{E_{23}} \dfrac{\partial W}{\partial X_2} \end{bmatrix}$$

$$M_{32} = \begin{bmatrix} X_2^{E_{32}} \dfrac{\partial W}{\partial X_3} & 0 & 0 \\ 0 & X_2^{E_{32}} \dfrac{\partial W}{\partial X_3} & 0 \\ 0 & E_{32} X^{E_{32}-1} \cdot W & X_2^{E_{32}} \dfrac{\partial W}{\partial X_3} \end{bmatrix}$$

$$M_{13} = \begin{bmatrix} X^{E_{13}} \dfrac{\partial W}{\partial X_1} & 0 & E_{13} X_3^{E_{13}-1} \cdot W \\ 0 & X_3^{E_{13}} \dfrac{\partial W}{\partial X_1} & 0 \\ 0 & 0 & X_3^{E_{13}} \dfrac{\partial W}{\partial X_1} \end{bmatrix}$$

$$M_{31} = \begin{bmatrix} X_1^{E_{31}} \dfrac{\partial W}{\partial X_3} & 0 & 0 \\ 0 & X_1^{E_{31}} \dfrac{\partial W}{\partial X_3} & 0 \\ E_{31} X_1^{E_{31}-1} \cdot W & 0 & X_1^{E_{31}} \dfrac{\partial W}{\partial X_3} \end{bmatrix}$$

This rather simple structure of $M_1, \ldots, M_{31}$ is one of the reasons for the particular choice of a flow $\vec{U}$ of the type (7.3).

3.3 Subroutine SORT2

For each individual tetrahedron, each one of the four linear 10×10 systems for the coefficients of the four basis functions is assembled according to (B22) and then solved by the function subroutine MATINV.* After that, the coefficients of their first and second partial derivatives are determined.

3.4 Function Subroutine QMLTI1

Because of the structure of $\vec{U}$ (see 7.3) and the basis functions $W_r(X)$, the matrix elements are linear combinations of integrals like:

$$\int_T X_1^{p_1} X_2^{p_2} X_3^{p_3} dX_1 dX_2 dX_3$$

$$= [X_{iL}-X_{i4}]_{det} \int_T \left[\sum_1^3 (X_{1L}-X_{14})\lambda_L+X_{14}\right]^{p_1} \left[\sum_1^3 (X_{2L}-X_{24})\lambda_L+X_{24}\right]^{p_2}$$

$$\times \left[\sum_1^3 (X_{3L}-X_{34})\lambda_L+X_{34}\right]^{p_3} d\lambda_1 d\lambda_2 d\lambda_3 \qquad \text{(B28)}$$

with integers p_1, p_2, p_3.

The right hand side is the transformation into barycentric coordinates (using (B11), (B12)). The determinant $[X_{iL}-X_{i4}]_{det}$ is evaluated in the main line and QMLTI1 evaluates the integral analytically for $p_1 + p_2 + p_3 \leq 6$** by use of Hammer's formula (B15).

---

*Supplied by its author D. Oracheski, Atmospheric Environment Service, Edmonton.

**A version for $p_1 + p_2 + p_3 \leq 10$ exists as well. It permits spatial dependencies in $\vec{U}$ upto power 6. However, this leads to prohibitively high CPU times.

This is the main reason for the particular structure chosen for $\vec{U}$: most spatial dependencies other than simple power laws would require numerical integrations or might have very little variability after discretization and integration. This would restrict the choice of possible bifurcation parameters.

QMLTI1 consists of six nested loops with four iterations each for the four summands $(X_{iL} - X_{i4})\lambda_L$, $i = 1,2,3$, X_{i4} in each one of the up to six factors in the integral of the right hand side of (B28). That means, the number of factors (i.e., $p_1 + p_2 + p_3$) is equal to the number of loops used and the remaining inner loops are skipped. The computation of the integrals over powers of the barycentric coordinates $\lambda_1, \lambda_2, \lambda_3$ (i.e, evaluation by Hammer's formula) is done in the innermost used loop only, and the outer loops are used only to assemble the correct linear combinations of these integrals with the $(X_{iL} - X_{i4})$, $L = 1,2,3$, X_{i4} as coefficients.

Whenever the main line enters a new tetrahedron, QMLTI1 evaluates all possible integrals satisfying $p_1 + p_2 + p_3 \leq 6$ on the right hand side of (B28). After the results have been multiplied with $[X_{iL} - X_{i4}]_{det}$, they are stored in the array QT and indexed according to their exponent combinations.* This method is considerably faster than recalculating these integrals whenever they are needed.

3.5 Function Subroutine QMLTI2

This routine uses the output of QMLTI1 and SORT2 to assemble the actual matrix elements. It consists of three nested loops with variable numbers of iterations (and variable initial value on the outer loop). This way, multiplications of up to three factors (i.e., nonlinear matrix elements W^{∇}_{onm}) consisting of polynomials with varying numbers of terms can be performed.

---

*This storage method requires an offset of one, i.e.: $\int_T X_1^{p_1} X_2^{p_2} X_3^{p_3}\, dX_1 dX_2 dX_3$ - QT (p_1+1, p_2+1, p_3+1), since a FORTRAN-66 array cannot be indexed by zero.

The proper selection of exponent combinations p_1, p_2, p_3 from the array QT is controlled by the data sets IEX, IEY, IEZ. The basis functions $W_r(X)$ are used throughout in the following order of terms:

$$W_r(X) = C(1) + C(2)X_1 + C(3)X_2 + C(4)X_3 + C(5)X_1^2 + C(6)X_2^2 + C(7)X_3^2 + C(8)X_1X_2 + C(9)X_2X_3 + C(10)X_1X_3 \tag{B29}$$

i.e., first coefficient: all exponents zero

second coefficient: X_1-exponent = 1, X_2, X_3-exponents = 0

⋮

tenth coefficient: X_1-exponent = X_2-exponent = 1,
X_3-exponent = 0.

This sequence of exponent combinations is kept for the first and second partial derivatives of $W_r(X)$ as well, where it goes up to fourth or first coefficient respectively only. IEX, IEY, IEZ now contain the partial sequences for the X_1, X_2, X_3 exponents which are in their combination the above sequence. The integers IPX, IPY, IPZ are either used to provide the required offset on QT (see last footnote) or to increase exponents according to spatial dependencies of $\vec{U}$ of powers higher than one.

3.6 Subroutine QMTRXB

QMTRXB assembles all linear 3×3 matrices on fluid space level with the structure (scalar) × (unit matrix). These are: W_{on}(LHS,(5.13)), W^{Δ}_{on} (viscosity term, (5.16)), $<W_{r_o}|M_1>$, $<W_{r_o}|M_2>$, $<W_{r_o}|M_3>$ (advective terms, (B28)). The subroutine works straightforwardly using QMLTI2 to calculate the scalar factor of these matrices and then determining the row and column numbers of these elements in the final $3N \times 3N$ matrices by the formula:

$$N = 3(I-1) + J \tag{B30}$$

with: N = row or column number in $3N \times 3N$ matrix

I = count number of node r_o (for rows) or r_n (for columns)

J = row or column number in 3×3 matrices on fluid space level.

The elements are indexed by these row and column numbers and stored accordingly in $3N \times 3N$ arrays. Since all diagonal matrix elements such as e.g., W_{nn} consist of a sum of two integrals over two tetrahedrons showing a face with r_n as centroid, whereas all nondiagonal elements consist of integrals over one tetrahedron only (as is evident from the fact that the support area for the basis functions $W_{r_n}(X)$ consists of two tetrahedrons as described above), the diagonal elements are calculated by the program in two different parts. To have these two parts properly added, the $3N \times 3N$ arrays are initialized to zero at the beginning and the matrix elements are added into them.

3.7 Subroutine QMTRXA

QMTRXA assembles and stores all linear 3 × 3 matrices on fluid space level with a structure different than the one of elements assembled by QMTRXB. These are all elements involving the advective matrices $M_{11}, \ldots, M_{31}$ listed in (B27). They can be written in a general form:

$$M_{ij} = X_j^{E_{ij}} \frac{\partial W}{\partial X_i} [I_3] + E_{ij} X_j^{E_{ij}-1} \cdot W[I_{ij}] \qquad \text{(B31)}$$

with $[I_{ij}]$ as a 3 × 3 matrix with $I_{nm} = 0$ for $n \neq 1;\ m \neq j$ and $I_{ij} = 1$.

The routine consists of two nested loops for the two indices i (outer loop: components of $\vec{U}$) and j (inner loop: spatial dependencies of $\vec{U}$). Again QMLTI2 is used to calculate the two matrix elements relating to the two different terms in (B31) separately and the exponents E_{ij} are user-supplied determining the spatial dependencies of the main flow $\vec{U}$ as specified in Equation (7.3). A formula of the type (B30) calculates rows and columns for the nondiagonal matrix elements $(i \neq j)$ of the second term in (B31) and they are added into arrays in the manner described in Section 3.6.

A third loop inside the two i,j-loops distributes all diagonal elements $(i = j)$ into storage arrays, according to a formula like (B30).

3.8 Subroutine QMTRXN

As is evident from the 3 × 9 subdecomposition of the nonlinear matrix elements in (5.12a), the nine-dimensional product vector

$V_{1j}(r_n)V_{1j}(r_m)$ can be contracted into a six-dimensional product vector for $n = m$, since equal components of $\vec{V}(r_n)$ become commuting and therefore:

$$a_1(V_{1i}(r_n)V_{1i}(r_n)) + a_2(V_{1i}(r_n)V_{1i}(r_n)) = (a_1 + a_2)(V_{1i}(r_n)V_{1i}(r_n)), \quad \text{(B32)}$$

and the 3×9 subdecomposition can be contracted into a 3×6 subdecomposition as well:

$$\begin{matrix} \frac{\partial W}{\partial X_1} & 0 & 0 & \frac{\partial W}{\partial X_2} & \frac{\partial W}{\partial X_3} & 0 \\ 0 & \frac{\partial W}{\partial X_2} & 0 & \frac{\partial W}{\partial X_1} & 0 & \frac{\partial W}{\partial X_3} \\ 0 & 0 & \frac{\partial W}{\partial X_3} & 0 & \frac{\partial W}{\partial X_1} & \frac{\partial W}{\partial X_2} \end{matrix} \quad \text{(B33)}$$

QMTRXN uses QMLTI 2 to calculate the three elements $< W_{r_o} | W_{r_n} | \frac{\partial W_{r_m}}{\partial X_i} >$, $i = 1,2,3$, and then assembles them into a subdecomposition according to (B33). Besides that, the three elements are stored temporarily, to be assembled in the main line later on into a 3×9 subdecomposition and written into a file, each with three indices determined by the nodal counts o,n,m in a formula of the type (B31). They are used later to assemble the matrix of the advective terms $(W^{\nabla}_{onm} + W^{\nabla}_{omn})$ in Equations (5.18), (5.19).

3.9 Calculation of the Total Column Number in the Nonlinear Matrix W^{∇}_{onm}

The column number N_c, say, of the nonlinear matrix W^{∇}_{onm} after subdecomposition can be calculated in the following way. First, there are two different types of interaction of nodes to distinguish:

1. self interaction: this relates to matrix elements like W_{onn}, where the node r_n interacts with itself. As has been shown by (B33), this leads to a six column subdecomposition or, equivalently, to six interactions between components.
2. mutual interaction: this relates to elements like W^{∇}_{onm}, $n \neq m$. If r_n and r_m are on different tetrahedrons only, W^{∇}_{onm} vanishes, as is evident by the disjoint support areas of the basis functions W_{r_n}, W_{r_m}. This means no interaction. If r_n and r_m are on one and the same tetrahedron, this evidently leads to a nine column subdecomposition or to nine interactions between components.

In a discretized rectangular domain, four types of tetrahedrons are possible.

1. Interior tetrahedrons with four counted nodes.
2. Tetrahedrons with one face in a surface plane of the domain. Three nodes are counted.
3. Tetrahedrons with two faces in surface planes, i.e., tetrahedrons sitting on the edges of the domain. Two nodes are counted.
4. Tetrahedrons with three faces in surface planes, i.e., tetrahedrons sitting in corners. Only one node is counted.

Clearly, the following interactions can be determined for these tetrahedrons:

Counted nodes	Interactions		Total Component
	Self	Mutual	
1	1	0	6
2	2	1	2 × 6 + 9 = 21
3	3	3	3 × 6 + 3 × 9 = 45
4	4	6	4 × 6 + 6 × 9 = 78

Moreover, the self interaction of a node shared by two tetrahedrons can only be counted once.

This clearly gives a formula for the total number of interactions in a given discretized domain or, equivalently, the total number of columns N_c, of its nonlinear matrix W^{∇}_{onm}:

$$N_c = T_1 \cdot 6 + T_2 \cdot 21 + T_3 \cdot 45 + T_4 \cdot 78 - I_f \cdot 6 \qquad \text{(B34)}$$

T_i = number of tetrahedrons with i counted nodes, $i = 1,2,3,4$

I_f = number of shared faces

3.10 Assembly of the Nonlinear Matrix W^{∇}_{onm} in the Main Line Program

The main line consists basically of five nested loops (see Figure 15) indexed by K,J,I,T,L1 in the order from outer loop to inner loop. K,J,I iterate through the discretized values of the Z,Y,Z-axes to generate the rectangular blocks, which split into five tetrahdrons each. T goes through these tetrahedrons and L1 through the four nodes per tetrahedron. Clearly, most of the operations have to be done within L1 (see Figure 16), as is the case for the assembly of W^{∇}_{onm}. This is done according to the structure of W^{∇}_{onm} as shown in Figure 14. The first 6N columns of the $3N \times N_c$ matrix are (partially) occupied by elements from self-interacting nodes only, and the remaining $(N_c - 6N)$ columns by the mutually interacting nodes. The sparseness of this matrix (as of the linear matrices) clearly depends on the level of discretization; the finer the discretization, the more tetrahedrons are used and therefore the sparser the matrix will be. Since the logic of the program can be followed by the diagrams of Figures 15 and 16, only a list of the FORTRAN symbols used in the program and in Figures 15, 16 as well is given here.

Arrays:

C(5,4), D(5,4), E(5,4) - X,Y,Z - coordinates for the vertices 1 to 4 of the tetrahedrons 1 to 5 per block.

SX(5), SY(5), SZ(5) - sums of the X,Y,Z-coordinates over the four vertices of the tetrahedrons 1 to 5 per block.

DET(5) - functional determinants between barycentric and cartesian coordinates for the tetrahedrons 1 to 5 per block.

XNC(4,4), YNC(4,4), ZNC(4,4) - X,Y,Z - coordinates of the centroids opposite to the faces 1 to 4 per tetrahedron 1 to 4 per block.

TNL(3) - array for temporary storage of the three elements $< W_{r_o} | W_{r_n} | \frac{W_{r_m}}{X_i} >$, i = 1,2,3 per nonlinear subdecomposition.

ANL(3,6,4,4) - array for temporary storage of the 3 × 6 subdecompositions stemming from the selfinteractions.

ANLF$(3N,N_c)$	- array holding the $3N \times N_c$ nonlinear matrix W^{∇}_{onm}.
IC1(N_c),IC2(N_c)	- arrays assigning the indices (IC1,IC2) of the appropriate velocity components in the product vector $(V_{1i}(r_n))$ to their column number between 1 and N_c.
NC(L3,T,I,J,K)	- array assigning the count number NC of a node to its identity (L3,T,I,J,K).
NCOL(4,4)	- array assigning the count number of combinations of a pair of nodes within a tetrahedron to the pair of their identity numbers (1...4,1...4) within this tetrahedrons. Only used for combinations on mutual interactions.

Loops:

I,J,K	- increments of X,Y,Z-coordinates.
T	- tetrahedrons 1 to 5 per block.
L1,L2,L3	- 3 nested loops, each one of them going through the four nodes per tetrahedron.
LL2,LL3	- 2 nested loops within L1, each one of them going through the four nodes per tetrahedron.

3.11 Routines ADV and TRNSP

The discretized versions (5.18), (5.19) for second and third perturbation differ from the one for the first perturbation by the term:

$$\sum_{r_n,r_m\in S} (W^{\nabla}_{onm} + W^{\nabla}_{omn})\vec{V}_i(r_m)\vec{V}_j(r_n), \quad r_o \in S \qquad \text{(B35)}$$

with indices i,j denoting perturbations.

Including the LHS-inverse denoted by W_{LH}^{-1} and introducing fluid space subdecomposition as well, this term can be rewritten:

$$W_{LH}^{-1} \sum_{n,m}^{3N} (W_{onm}^{\nabla} + W_{omn}^{\nabla}) V_{im} V_{jn} = W_{LH}^{-1} \sum_{n}^{3N} \left[\sum_{m}^{3N} \left((W_{on}^{\nabla})_m + \overline{(W_{on}^{\nabla})_m} \right) V_{im} \right] V_{jn}$$

$$0 = 1,\ldots,3N \qquad \text{(B36)}$$

with the indices o,n,m going from 1 to 3N over all 3N velocity components V_{im}, V_{jn} through the N points of S.

Moreover:

$$\left.\begin{array}{l} (W_{on}^{\nabla})_m = W_{onm}^{\nabla} : \\ \overline{(W_{on}^{\nabla})_m} = W_{omn}^{\nabla} : \end{array}\right\} \text{3N} \times \text{3N matrix indexed by } m$$

(From the definition (5.17) of W_{onm}^{∇} it is clear that: $(W_{on}^{\nabla})_m \neq \overline{(W_{on}^{\nabla})_m}$. Clearly, the two sets are not identical since $W_{onm}^{\nabla} \neq \overline{W_{omn}^{\nabla}}$.)

This reveals the term in the square brackets of (B36) as a linear combination of matrices and W_{LH}^{-1} can therefore be moved into the sum:

$$\sum_{n}^{3N} \left[\sum_{m}^{3N} \left(W_{LH}^{-1} \left((W_{on}^{\nabla})_m + \overline{(W_{on}^{\nabla})_m} \right) \right) V_{im} \right] V_{jn} \qquad \text{(B37)}$$

In the subroutine ADV, the elements of the subdecomposed W_{onm}^{∇}, as calculated by QMTRXN and stored with their three indices, are sorted once after the index n and once after the index m. This gives all 3N subdecompositions of $(W_{on}^{\nabla})_m$ and $\overline{(W_{on}^{\nabla})_m}$. After this sorting process, the pairs $(W^{\nabla}_{on})_m$, $\overline{(W^{\nabla}_{on})}_m$ are added and multiplied by W^{-1}_{LH}. A transposing routine rearranges the elements of the resulting 3N matrices in the sequence in which they are being used to assemble and evaluate the RHS of (B36) in the programs which evaluate Equations (5.18) and (5.19). The same type of transposing routine (called TRNSP) is used to rearrange the elements of all linear matrices into the sequence as they flow into routines used later to evaluate Equation (5.19) or (5.20) respectively.

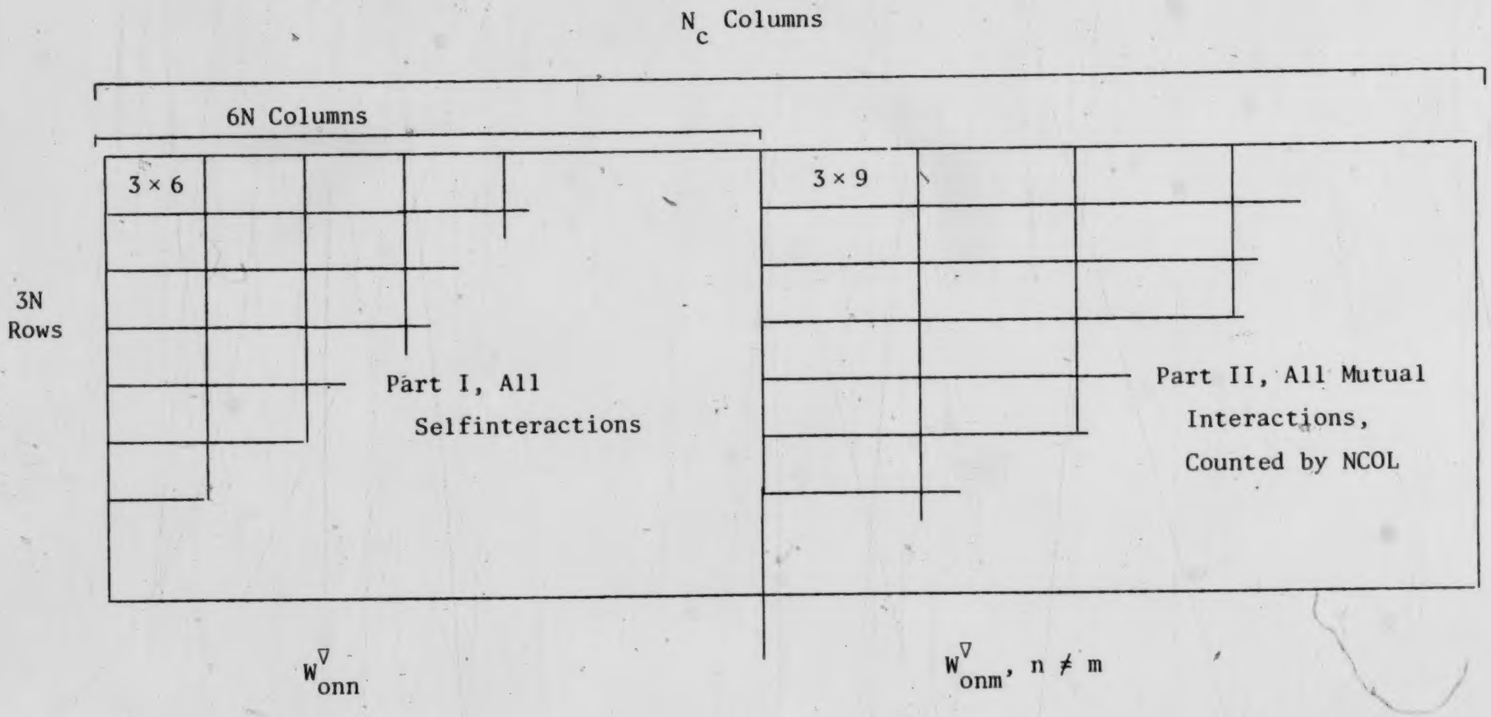
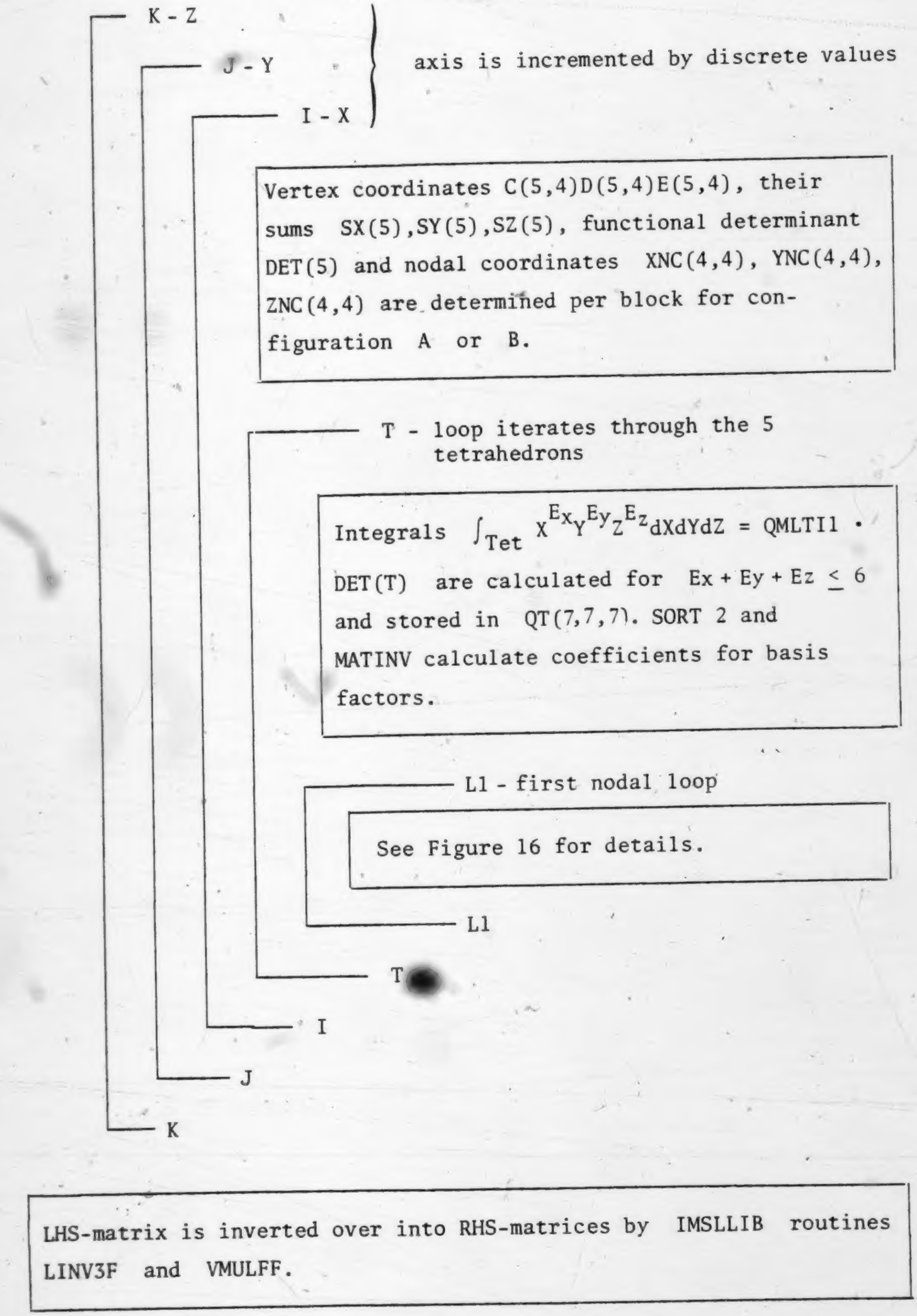

N_c Columns
6N Columns
3N Rows
3 × 6
Part I, All Selfinteractions
3 × 9
Part II, All Mutual Interactions, Counted by NCOL
W^{∇}_{onn}
W^{∇}_{onm}, $n \neq m$

Figure 14. Structure of Nonlinear Matrix W^{∇}_{onm}



Vertex coordinates C(5,4)D(5,4)E(5,4), their sums SX(5),SY(5),SZ(5), functional determinant DET(5) and nodal coordinates XNC(4,4), YNC(4,4), ZNC(4,4) are determined per block for configuration A or B.

T - loop iterates through the 5 tetrahedrons

Integrals $\int_{Tet} X^{E_x} Y^{E_y} Z^{E_z} dXdYdZ$ = QMLTI1 · DET(T) are calculated for $Ex + Ey + Ez \leq 6$ and stored in QT(7,7,7). SORT 2 and MATINV calculate coefficients for basis factors.

L1 - first nodal loop

See Figure 16 for details.

LHS-matrix is inverted over into RHS-matrices by IMSLLIB routines LINV3F and VMULFF.

Figure 15. Outer Loops of DSCRB

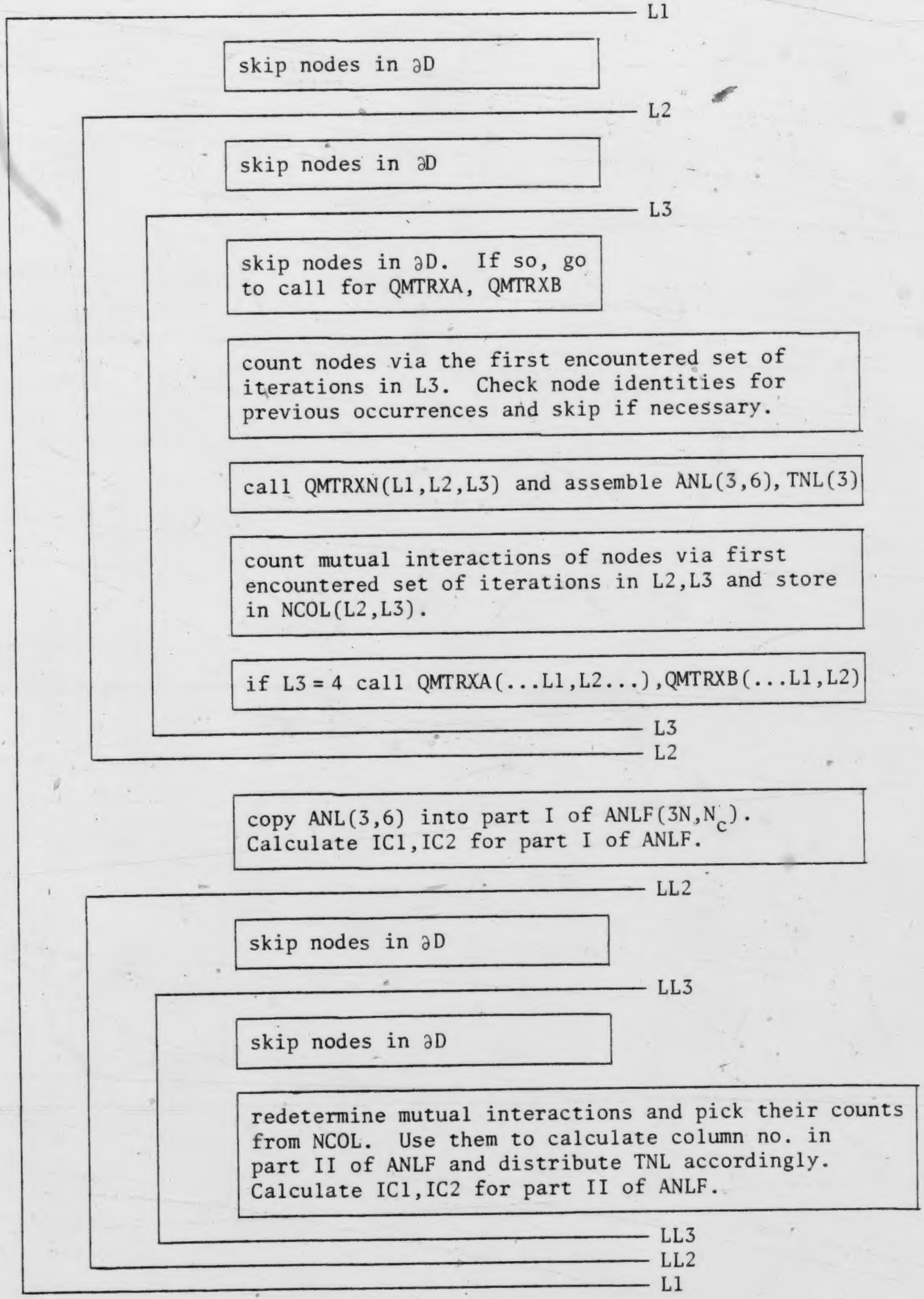

L1
skip nodes in ∂D
L2
skip nodes in ∂D
L3
skip nodes in ∂D. If so, go to call for QMTRXA, QMTRXB
count nodes via the first encountered set of iterations in L3. Check node identities for previous occurrences and skip if necessary.
call QMTRXN(L1,L2,L3) and assemble ANL(3,6), TNL(3)
count mutual interactions of nodes via first encountered set of iterations in L2,L3 and store in NCOL(L2,L3).
if L3 = 4 call QMTRXA(...L1,L2...),QMTRXB(...L1,L2)
L3
L2
copy ANL(3,6) into part I of ANLF(3N,N_c). Calculate IC1,IC2 for part I of ANLF.
LL2
skip nodes in ∂D
LL3
skip nodes in ∂D
redetermine mutual interactions and pick their counts from NCOL. Use them to calculate column no. in part II of ANLF and distribute TNL accordingly. Calculate IC1,IC2 for part II of ANLF.
LL3
LL2
L1

Figure 16. Inner Loops of DSCRB

APPENDIX C

1. FLOWS TESTED FOR BIFURCATIONS

All flows show the main characteristics already mentioned in the text: linear shear flows or one component flows undergo stationary bifurcations only; $\eta_c \sim |\nu|$, $\omega_o \sim |\nu|$, $\eta_2 \sim |\nu^{-1}|$, $\mu_2 \sim |\nu^{-1}|$; qualitatively correct behaviour is found in almost all cases by loss of stability with increasing shear. All flows were unstable for positive viscosities except flow Number 3) which became critical for $\nu = 1$ at:

$$U_x = 5\,, \quad U_z = 98.46y\,,$$

for $\nu = .1$ at:

$$U_x = 5., \quad U_z = 48.98y$$

and for $\nu = .01$ at:

$$U_x = 5., \quad U_z = 44.99y$$

$$U_x = 1., \quad U_z = 9.35y\,.$$

In all four cases the shear coefficient of U_z was used as bifurcation parameter and the stationary bifurcations went from unstable to stable with increasing shear.

Flows with Linear Shear

1) $U_y = C_{yx}X$

2) $U_x = C_x\,, \quad U_y = C_{yx}X$

3) $U_x = C_x\,, \quad U_z = C_{zy}Y$

4) $U_x = C_{xx}X\,, \quad U_y = C_{yy}Y\,, \quad U_z = C_z$

5) $U_x = C_{xx}X$, $U_y = C_{yy}Y$, $U_z = C_{zz}Z$

Only stationary bifurcations could be found, which caused a loss of stability by increasing the shear coefficients above their critical value. These occur e.g. for viscosity $\nu = -1$ and for Flow 1) at $C_{yx} = 6.56$, Flow 2) at $C_x = .3$, $C_{yx} = 7.1$ and Flow 3) at $C_x = 10$, $C_{zy} = 52.8$. The critical values of the shear coefficients show only a weak dependence on the shear-free components such as C_x in Flow 2) and 3) and C_z in Flow 4).

For the Flows 4) and 5) the shear coefficients were set as linear functions of a bifurcation parameter η and the occurrence and value of critical shear coefficients varied with the combination of proportionality constants used. The critical shear coefficients however stayed within the orders of magnitude of 10 or 1.

One-Component Flows with Nonlinear Shear

6) $U_x = C_{xy}Y^2$

7) $U_x = C_{xz}Z^3$

Again only stationary bifurcations could be found. They occurred for viscosity $\nu = -1$ and Flow 6) at $C_{xy} = 8.07$ and Flow 7) at $C_{xz} = 9.39$.

Two-Component Flows with Nonlinear Shear

8) $U_x = C_{xy}Y^2$, $U_z = C_z$

9) $U_x = C_{xz}Z^3$, $U_y = C_y$

10) $U_x = C_x - C_{xy}Y^2$, $U_z = C_z$

All three flows show unstable Hopf bifucations (besides stationary ones) for shear coefficients between zero and one for flows 9) and 10) and around five for flow 8) (Viscosity $\nu = -1.$) They are subcritical for flow 9), supercritical for flow 10) and both for flow 8). Flow 8) also showed the first stable (subcritical) Hopf bifurcation found around: $\nu = -2.$, $C_z = -11.$, $C_{xy} = 9.$.

Three Component Flows with Nonlinear Shear

11) $U_x = C_{xy}Y^2$, $U_y = C_{yx}X$, $U_z = C_z$

12) $U_x = C_{xy}Y^2$, $U_y = C_{yx}X^3$, $U_z = C_z$

13) $U_x = C_{xz}Z$, $U_y = C_{yx}X^3$, $U_z = C_{zy}Y^3$

14) $U_x = C_{xz}Z^3$, $U_y = C_{yx}X^3$, $U_z = C_{zy}Y^3$

15) $U_x = C_{xy}Y^2$, $U_y = C_{xz}Z^2$, $U_z = C_{zx}X^2$

16) $U_x = C_{xz}Z^4$, $U_y = C_{yx}X^4$, $U_z = C_{zy}Y^4$ * .

17) $U_x = C_{xz}Z^3 + C_{xy}Y^3$, $U_y = C_{yx}X^3 + C_{yz}Z^3$, $U_z = C_{zy}Y^3 + C_{zx}X^3$

18) $U_x = C_{xy}Y^2 + C_{xz}Z^3$, $U_y = C_{yx}X^2 + C_{yz}Z^2$, $U_z = C_{zx}X^2 + C_{zy}Y^3$

For all flows Hopf bifurcations were quite frequent. In general, the critical values for the coefficients stayed in their order of magnitude between one and ten, irrespective whether some of them were kept fixed (bifurcations on flows with some of their coefficients held fixed at other orders of magnitude were rather sparse) or all were linear functions of a bifurcation parameter η. Also secondary bifurcations

---

*An experimental version of the discretization program extended to spatial dependencies up to power 6 was used here.

for a time setting of $t = 0$ on the first orbit were found. All possible types and combinations of criticality and stability could be observed. However, supercritical and stable bifurcations appeared more frequent than others.

Flows with Affine and Power Dependencies on the Bifurcation Parameter η

19) $U_x = (C_{xy} \cdot \eta + C_1)Y^2, \quad U_y = (C_{yz} \cdot \eta + C_2)Z^2, \quad U_z = (C_{zx} \cdot \eta + C_3)X^2$

20) $U_x = (C_{xy} \cdot \eta^{e_1} + C_1)Y^2, \quad U_y = (C_{yz} \cdot \eta^{e_2} + C_2)Z^2,$

$U_z = (C_{zx} \cdot \eta^{e_3} + C_3)x^2$

21) $U_x = \eta(C_x + C_{xy}Y^2), \quad U_y = \eta(C_y + C_{yz} \cdot Z^2), \quad U_z = \eta(C_z + C_{zx} \cdot X^2)$

22) $U_x = (C_{xy} \cdot \eta^{e_1} + C_1)Y^2 + (C_{xz}\eta^{e_2} + C_2)Z^3$

$U_y = (C_{yx} \cdot \eta^{e_3} + C_3)X^3 + (C_{yz}\eta^{e_4} + C_4)Z^2$

$U_z = (C_{zx} \cdot \eta^{e_5} + C_5)X^2 + (C_{zy}\eta^{e_6} + C_6)Y^3$

23) $U_x = C_1 + C_x\eta + (C_4 + C_{xy}\eta)Y^2$

$U_y = C_2 + C_y\eta + (C_5 + C_{yz}\eta)Z^2$

$U_z = C_3 + C_z\eta + (C_6 + C_{zx}\eta)X^2$

The findings for this group are in principle the same as for the previous one with linear dependencies only. However, stable and supercritical Hopf bifurcations were more frequent and less sensitive against variation of parameters, coefficients, and exponents in the affine and power dependencies. Besides, third level bifurcations (time settings on first and secondary orbit were $t = 0$) could be found only in this group.

APPENDIX D

INTRODUCTION TO NUMERICAL WORK ON THE NS-EQUATION

Most of the numerical work done so far on the NS-equations is restricted to two dimensional flows, since there the NS-equations can be replaced by a so-called stream function-vorticity formulation.

Applying the curl operator on the NS-equation gives the vorticity equation:

$$\frac{\partial \vec{\omega}}{\partial t} + \vec{U} \cdot \vec{\nabla} \vec{\omega} + \vec{\omega} \cdot \vec{\nabla} \vec{U} = \nu \nabla^2 \vec{\omega} , \qquad \text{(A)}$$

$$\text{(with: } \vec{\omega} = \vec{\nabla} \times \vec{U} \text{ as vorticity)} \qquad \text{(A1)}$$

and eliminates the pressure gradient since $\overline{\nabla} \times \overline{\nabla} \equiv 0$. In a two dimensional flow field the vorticity vector reduces to one component, ω_Z, say perpendicular to the plane spanned by X,Y of the flow and the vorticity equation can be written as a scalar equation:

$$\frac{\partial \omega_Z}{\partial t} = -U_x \frac{\partial \omega_Z}{\partial X} - U_y \frac{\partial \omega_Z}{\partial Y} + \nu \left(\frac{\partial^2 \omega_Z}{\partial X^2} + \frac{\partial^2 \omega_Z}{\partial Y^2} \right) \qquad \text{(B)}$$

$$\text{with: } \omega_Z = \frac{\partial U_Y}{\partial X} - \frac{\partial U_X}{\partial Y} \qquad \text{(B1)}$$

Also, in two dimensions, the incompressibility condition:

$$\text{div } \vec{U} = 0 \qquad \text{(C)}$$

reduces to:

$$\frac{\partial U_x}{\partial X} + \frac{\partial U_y}{\partial Y} = 0, \qquad \text{(C1)}$$

which can always be satisfied by introducing the (scalar) stream function ψ as:

$$U_x = \frac{\partial\psi}{\partial Y} ; \quad U_y = - \frac{\partial\psi}{\partial X} \tag{D}$$

Clearly, in a stationary flow the following relation holds along the particle trajectories:

$$\frac{U_x}{U_y} = \frac{dX}{dY} \Longrightarrow -U_y dX + U_x dY = 0 \tag{E}$$

and the isolines of the stream function (i.e. the streamlines) must be identical to the trajectories since

$$d\psi = \frac{\partial\psi}{\partial X} dX + \frac{\partial\psi}{\partial Y} dY = -U_y dX + U_x dY \tag{F}$$

Combining the two-dimensional vorticity (B1) with the stream function (D) gives the elliptic Poisson equation:

$$\frac{\partial^2\psi}{\partial X^2} + \frac{\partial^2\psi}{\partial Y^2} = -\omega_z \tag{G}$$

Equations (B), (D), (G) are four equations for the four unknowns U_x, U_y, ψ, ω_z. They are mostly solved in engineering applications under the appropriate boundary conditions by finite-difference or finite element methods. Their advantages as well as their disadvantages are evident; pressure is eliminated and the incompressibility condition is incorporated most conveniently steady-state flow patterns can be obtained according to relations (E), (F) most easily by graphing the isolines of the stream function ψ.

On the other hand, however, the stream function concept cannot be applied in three dimensional flows and time-dependent solutions would require substantial modifications to introduce evolution equations for U_x, U_y, and ψ as well. Since, in the present context, a monitoring of the evolution time of a velocity vector through a particular point in the fluid domain is required, this vorticity-stream function concept cannot be applied.

The publications on the stream function-vorticity approach are numerous. See e.g. Morris [15], Smith, Jr. and Kidd [16], Rubin and Graves, Jr. [17,18], Boney [19], Hefner [20], Hirsh [21], Suttles [22], Zoby [23] for treatment and description of various technical and computational details, the associated problems of convergence and the numerical algorithms applied to achieve convergence. Most of the work deals with the so-called cavity problem, where the fluid is considered enclosed in a rectangular cavity and its steady motion is driven by one sliding wall or more via non slip boundary conditions.

In the seventies, finite-element techniques in connection with the Galerkin method were adapted to solve the NS-equations. However, the amount of numerical operations and storage requirements are considerably higher than in e.g. finite-difference methods, since the coefficient matrices contain more nonzero entries. Representative papers would be, e.g., Cheng [24], Taylor and Hood [25], Kawahara et. al. [26], Fortin and Thomasset [27], Olson and Tuann [28].

Again most of the work done so far is restricted to two dimensions. One of the reasons for this seems to be that the amount of numerical computations and storage requirements increases by almost one order of magnitude from two to three dimensions, as could be concluded in the present work. Moreover, the finite-element technique depends on a triangulation (i.e., partition into triangles) of the two-dimensional fluid domain, which naturally has to be based on tetrahedrons in three dimensions instead of triangles. The problems involved therewith do not permit an almost straight-forward interpolation from two to three dimensions, as is the case for rectangular finite differences. However, application of the finite-element/Galerkin technique to three dimensions seems at present the only feasible method to discretize the NS-equation for the problem at hand into a system of evolution ODE's which can be solved without algorithms for correction to satisfy incompressibility in between two timesteps, as is important for the applicability of the program BIFOR 2, which tests the ODE-system for Hopf bifurcations. The main reference for the analytic background on which the here employed techniques are based, as well as for a very thorough and complete extensive analysis of the NS-equations is the book by Temam [12].

APPENDIX E

1. THE INERTIAL SUBRANGE LAW

(a) Kolmogorov's Derivation [13], 1941

Kolmogorov (13) divided the spectrum of energy per unit mass versus wave number of an isotropic homogeneous turbulent flow into three ranges:

1. the production range at low wave numbers, which draws energy from the mean flow and converts it into turbulent fluctuations;
2. the inertial range at intermediate wave numbers, in which energy is shifted upwards towards higher wave numbers without much production or dissipation taking place;
3. the dissipation range at high wave numbers, in which the energy put into the production range is finally dissipated into a motion comparable to Brownian motion.

Assuming that the transfer of energy from production range through inertial range into dissipation range is a local phenomenon, these three ranges are virtually independent of each other. In particular, the spectra in the inertial and dissipation ranges will not be much affected by the precise way in which energy is put into the production range. Moreover, in a stationary mean flow the production rate of turbulent energy must be equal to the dissipation rate, ϵ, say.

Calling the spectral density $E(k)$ and recognizing it as energy per unit mass within the wave number range $(k, k+dk)$, it must have dimension L^3T^{-2}. The dissipation rate ϵ, as energy per unit mass per time, must have dimension L^2T^{-3}. It is also clear that smaller eddies (i.e. larger k) contain less energy in their associated wave number range. One can therefore write:

$$E(k) = C_k \cdot f_k(k) \downarrow k, \tag{E1}$$

i.e., $E(k)$ decreases monotonically with increasing wave number k.

On the other hand, a higher dissipation rate ϵ means that more energy has to pass through from k to $k + dk$. That means:

$$E(k) = C_\epsilon \cdot f_\epsilon(\epsilon) \uparrow \epsilon, \tag{E2}$$

i.e., $E(k)$ increases monotonically with increasing dissipation rate ϵ.

It is also evident that in the limit for laminar flow $\epsilon \rightarrow 0$:

$$E(k) = 0 \quad \text{for} \quad \epsilon = 0 \tag{E3}$$

Since dissipation in the inertial subrange is negligible, $E(k)$ cannot depend on the viscosity there. Therefore the only possible dependent parameters left are ϵ and k. $E(k)$ must therefore be a combination of (E1) and (E2) under consideration of (E3), which yields:

$$E(k) = C \cdot f_\epsilon(\epsilon) \cdot f_k(k) \tag{E4}$$

the functions $f_\epsilon(\epsilon)$ and $f_k(k)$ can now be easily dimensionally scaled as:

$$f_\epsilon(\epsilon) = \epsilon^{2/3}; \quad f_k(k) = k^{-5/3} \tag{E5}$$

and C as a dimensionless constant to obtain the correct dimension

L^3T^{-2} for E(k) and observing L^2T^{-3} and L^{-1} as dimensions for ϵ and k respectively.

(E4) and (E5) give now the well-known inertial subrange or (-5/3)-law:

$$E(k) = C \cdot \epsilon^{2/3} k^{-5/3} \tag{E6}$$

b) Heisenberg's Derivation [14], 1948

This derivation is directly related to the finite closure methods. Beginning with the NS-equation

$$\frac{\partial}{\partial t} U_i(\vec{X}) + \frac{\partial}{\partial X_{1j}} [U_i(\vec{X}_1)U_j(\vec{X}_1)] = \nu\nabla_1^2 U_i(\vec{X}_1) - \frac{\partial p(\vec{X}_1)}{\partial X_{1j}} \qquad \text{(E7)*}$$

where the incompressibility condition

$$\frac{\partial}{\partial X_{1j}} U_j(\vec{X}_1) = 0 \tag{E8}$$

is used in the advective term, the equation (E7) is multiplied with $U_k(\vec{X}_2)$ and ensemble averages are taken:

$$\begin{aligned}\frac{\partial}{\partial t} \overline{U_i(\vec{X}_1)U_k(\vec{X}_2)} &+ \frac{\partial}{\partial X_{1j}} \overline{U_i(\vec{X}_1)U_j(\vec{X}_1)U_k(\vec{X}_2)} \\ &= \nu\nabla_1^2 \overline{U_i(\vec{X}_1)U_k(\vec{X}_2)} - \frac{\partial}{\partial X_{1i}} \overline{p(\vec{X}_1)U_k(\vec{X}_2)}\end{aligned} \tag{E9}$$

Since homogeneity is assumed, the averaged products depend on the spatial difference $\vec{r} \equiv \vec{X}_2 - \vec{X}_1$ only. Defining the correlations:

---

*For reasons of conciseness, time dependence is not written explicitely. Subscripted numbers denote points and letters denote components.

$$R_{ij}(\vec{r}) \equiv \overline{U_i(\vec{X}_1)U_j\vec{X}_2} \qquad \text{(E10)}$$

$$S_{ijk}(0,\vec{r}) \equiv S_{ijk}(\vec{r}) \equiv \overline{U_i(\vec{X}_1)U_j(\vec{X}_1)U_k(\vec{X}_2)} \qquad \text{(E11)}$$

$$P_j(\vec{r}) \equiv \overline{p(\vec{X}_1)U_j(\vec{X}_2)} \qquad \text{(E12)}$$

and using $\frac{\partial}{\partial r_i} = -\frac{\partial}{\partial X_{1i}}$, Equation (E9) writes as:

$$\frac{\partial}{\partial t} R_{ik}(\vec{r}) - \frac{\partial}{\partial r_j} S_{ijk}(\vec{r}) = \nu\nabla^2_{\vec{r}} R_{ik}(\vec{r}) + \frac{\partial}{\partial r_i} P_k(\vec{r}) \qquad \text{(E13)}$$

In the same way, the following conditions can be derived straight forwardly from the incompressibility condition (E8):

$$\frac{\partial}{\partial r_i} R_{ik}(\vec{r}) = \frac{\partial}{\partial r_k} S_{ijk}(\vec{r}) = \frac{\partial}{\partial r_i} P_i(\vec{r}) = 0 \qquad \text{(E14)}$$

Fourier-transforming the correlations (E10), (E11), (E12):

$$\phi_{ij}(\vec{k}) = (2\pi)^{-3} \int R_{ij}(\vec{r})e^{-i\vec{k}\vec{r}}\, d\text{ Vol} \qquad \text{(E15)}$$

$$\gamma_{ijk}(\vec{k}) = (2\pi)^{-3} \int S_{ijk}(\vec{r})e^{-i\vec{k}\vec{r}}\, d\text{ Vol} \qquad \text{(E16)}$$

$$\theta_i(\vec{k}) = (2\pi)^{-3} \int P_i(\vec{r})e^{-i\vec{k}\vec{r}}\, d\text{ Vol}, \qquad \text{(E17)}$$

gives for Equations (E13), (E14):

$$\frac{\partial}{\partial t} \phi_{ik}(\vec{k}) - ik_j\gamma_{ijk}(\vec{k}) = -\nu k^2\phi_{ik}(\vec{k}) + ik_i\theta_k(\vec{k}) \qquad \text{(E18)}$$

$$k_i\phi_{ik} = k_k\gamma_{ijk} = k_i\theta_i = 0 \qquad \text{(E19)}$$

Besides homogeneity, another important simplification is isotropy. The correlation tensors ϕ_{ij}, γ_{ijk}, θ_i therefore take the following form:

$$\phi_{ik}(\vec{k}) = a_1(k)\delta_{ik} + a_2(k)k_k k_k \tag{E20}$$

$$\gamma_{ijk}(\vec{k}) = b_1(k)k_i k_j k_k + b_2(k)k_i\delta_{jk} + b_3(k)k_j\delta_{ik} + b_4(k)\delta_{ij} \tag{E21}$$

In the tensor context isotropy is synonymous with invariance under the full rotation group, ie.,:

$$\phi_{ik} = \phi'_{ik} = a_{ji}a_{Lk}\phi_{jL} \tag{E22}$$

$$\gamma_{ijk} = \gamma'_{ijk} = a_{Li}a_{mj}a_{nk}\gamma_{Lmn}, \tag{E23}$$

with a_{ij} as the matrix performing the rotation of coordinates and the prime denoting a rotated coordinate system. The isotropy conditions (E22), (E23) can now easily be verified for the structures (E20) and (E21) respectively, by using linearity of the rotation and isotropy of δ_{ij}. Also the following symmetry is evident:

$$S_{ijk} = S_{jik} \Longrightarrow b_2 = b_3 \tag{E24}$$

The pressure correlation θ_j vanishes

$$\theta_j(k) = 0, \tag{E25}$$

as is evident by physical arguments, since under isotropy pressure cannot correlate with a particular direction such as the velocity vector $\vec{U}$.

The incompressibility conditions (E19) simplify the correlation tensors ϕ_{ij}, γ_{ijk} further, since they require:

$$a_1(k) + k^2 a_2(k) = 0 \tag{E26}$$

$$k^2 b_1(k) + 2b_2(k) = b_4(k) = 0, \tag{E27}$$

And ϕ_{ij}, γ_{ijk} can now be expressed by only one scalar function each:

$$\phi_{ij}(\vec{k}) = \frac{E(k)}{4\pi k^2}\,(k^2\delta_{ik} - k_i k_k) \tag{E28}$$

$$\gamma_{ijk}(\vec{k}) = i\gamma(k)\left(k_i k_j k_k - \frac{1}{2}\,k^2(k_i\delta_{jk} + k_j\delta_{ik})\right) \tag{E29}$$

Entering (E25), (E28), (E29) into (E18) gives:

$$\frac{\partial}{\partial t}\,E(k) + \nu k^2 E(k) = 2\pi k^6 \gamma(k) \tag{E30}$$

Forming the trace of Equation (E28) leads to:

$$E(k) = 2\pi k^2 \phi_{ii}(\vec{k}) \tag{E31}$$

On the other hand, the kinetic energy density of the flow ϵ_{kin} is:

$$\begin{aligned}\epsilon_{kin} &= \frac{\rho}{2}\,R_{ii}(0) = \frac{\rho}{2}\,\overline{U_i(\vec{X}_1)U_i(\vec{X}_1)} \\ &= \frac{\rho}{2}\int \phi_{ii}(\vec{k})e^{i\vec{k}\vec{r}}\,d^3\vec{k}\Big|_{\vec{r}=0} = \frac{\rho}{2}\int \phi_{ii}(\vec{k})d^3\vec{k}\,,\end{aligned} \tag{E32}$$

and substituting (E31) into (E32) gives:

$$\epsilon_{kin} = \rho V \int_0^\infty E(k)dk \tag{E33}$$

$E(k)$ must be therefore the spectral density of energy.

Next, one observes

$$\int k_j \gamma_{iji}(\vec{k}) d^3\vec{k} = 0 \tag{E34}$$

This can easily be derived from the Fourier-transform:

$$i \int k_j \gamma_{iji} e^{i\vec{k}\vec{r}} d^3\vec{k} = \frac{\partial}{\partial r_j} S_{iji}(\vec{r}) \tag{E35}$$

$$= \frac{\partial}{\partial r_j} \overline{U_i(\vec{X}_1)U_j(\vec{X}_1)U_i(\vec{X}_1+\vec{r})} = \overline{U_i(\vec{X}_1)U_j(\vec{X}_1) \frac{\partial}{\partial X_{1j}} U_i(\vec{X}_1+\vec{r})},$$

For $\vec{r} = 0$ and the incompressibility condition

$$\frac{\partial U_j(\vec{X}_1)}{\partial X_{1j}} = 0$$

this last term gives

$$\frac{1}{2} \overline{\frac{\partial}{\partial X_{1j}} (U_i(\vec{X}_1)U_j(\vec{X}_1)U_i(\vec{X}_1))} = \frac{1}{2} \frac{\partial}{\partial X_{1j}} \overline{U_i(\vec{X}_1)U_j(\vec{X}_1)U_i(\vec{X}_1)} = 0, \tag{E36}$$

since the differentiation takes place on a constant value.

Integrating Equation (E30) over k-space therefore gives:

$$\frac{\partial}{\partial t} \int_0^\infty E(k)dk + \nu \int_0^\infty k^2 E(k)dk = 2\pi \int_0^\infty k^6 \gamma(k)dk = 0 \tag{E37}$$

Heisenberg's assumption is now that the loss of kinetic energy of small eddies (i.e., large wave numbers) is negligible compared to the loss on large eddies (i.e., small wave numbers). That is, under use of (E33);

$$\frac{\partial}{\partial t}\epsilon_{kin}\ (k = \text{large up to } \infty) = \rho V \frac{\partial}{\partial t}\int_k^\infty E(k')dk' = 0 \qquad \text{(E38)}$$

Integrating (E30) from 0 to k and from k to ∞:

$$\frac{\partial}{\partial t}\int_0^k E(k')dk' + \nu\int_0^k k'^2E(k')dk' = 2\pi\int_0^k k'^6\gamma(k')dk' \qquad \text{(E39)}$$

$$\frac{\partial}{\partial t}\int_k^\infty E(k')dk' + \nu\int_k^\infty k'^2E(k')dk' = 2\pi\int_k^\infty k'^6\gamma(k')dk' \qquad \text{(E40)}$$

Following the finite-closure method, the third-order correlation term with $\gamma(k)$ in (E39) is now to be replaced by a second-order correlation term with $E(k)$. A relation between these two terms is given by (E40) and (E38), however, for large wave numbers only.

Based on the viscosity term of (E39), a turbulence viscosity $\bar{\eta}(k)$ can be introduced:

$$-\nu\int_k^\infty k'^2E(k')dk' = \bar{\eta}(k)\int_0^k k'^2E(k')dk' \qquad \text{(E41)}$$

recalling from (E37):

$$-2\pi\int_0^k k'^6\gamma(k')dk' = 2\pi\int_k^\infty k'^6\gamma(k')dk', \qquad \text{(E42)}$$

this relation (E42) together with (E41) and (E38) can be used in (E40) to obtain:

$$2\pi\int_0^k k'^6\gamma(k')dk' = \bar{\eta}(k)\int_0^k k'^2E(k')dk', \qquad \text{(E43)}$$

a relation between second and third-order correlations expressed by $E(k)$ and $\gamma(k)$, respectively, for small wave numbers. Equation (E39) now becomes:

$$\frac{\partial}{\partial t}\int_0^k E(k')dk' = -(\bar{\eta}(k) + \nu)\int_0^k k'^2 E(k')dk' \qquad \text{(E44)}$$

From the kinetic theory of gases, the following proportionality between viscosity ν and the kinetic energy per unit mass is known:

$$\nu \sim (kmT)^{\frac{1}{2}} \qquad \text{(E45)}$$

with kmT as kinetic energy per unit mass determined by: k = Boltzmann constant, m = molecular mass, T = temperature.

This relation between energy and viscosity can now be used to express the turbulent viscosity $\bar{\eta}(k)$ in terms of spectral energy density $E(k)$. Establishing, as in Kolmogorov's derivation, the dimensions of the involved quantities:

spectral energy density:	$E(k)$	dimension: S^3t^{-2}
wave number:	k	S^{-1}
viscosity:	$\bar{\eta}$	S^2t^{-1}
energy per unit mass:	$\int E(k)dk$	S^2t^{-2},

the turbulent viscosity can be dimensionally scaled by use of the proportionality (E45) as:

$$\bar{\eta}(k) = \int_k^\infty \eta(k')dk' \qquad \text{(E46)}$$

$$\eta(k) = \alpha \left(\frac{E(k)}{k^3} \right)^{\frac{1}{2}} \qquad \text{(E47)}$$

where α is a dimension less constant.

Equation (E44) becomes now:

$$\frac{\partial}{\partial t} \int_0^k E(k')dk' \equiv \overline{\varepsilon} = \left[\int_k^\infty {\frac{E(k')}{k'^3}}^{\frac{1}{2}} dk' + \nu \right] \int_0^k k'^2 E(k')dk' \qquad \text{(E48)}$$

Assuming time intervals, where the loss of energy of large eddies, i.e., $\overline{\varepsilon}$, can be considered constant, this equation is solved by:

$$E(k) = \frac{C(\overline{\varepsilon})^{2/3} k^{-5/3}}{\left[1 + \frac{8}{3\alpha^2} (k/k_j)^4 \right]^{4/3}} \qquad \text{(E49)}$$

with: $k_j = \left(\frac{\overline{\varepsilon}}{\nu^3} \right)^{\frac{1}{4}}$.

Therefore:

$$E(k) \sim k^{-5/3} \quad \text{for} \quad k \ll k_j$$

$$E(k) \sim k^{-7} \quad \text{for} \quad k \gg k_j$$

Besides the inertial subrange law for $k \ll k_j$ Heisenbergs derivation gives a power law for the dissipative range $k \gg k_j$ as well. This is possible, due to the use of kinetic gas theory in the derivation.

2. THE TAYLOR HYPOTHESIS

Measurements, which are to be used to determine the spectrum $E(k)$ introduced in Part 1 must consist of a series of measuring devices distributed one-dimensionally and equi-spaced with a certain distance d parallel to the mean steady and homogeneous flow $\overline{U}$. Moreover, this set of measuring devices should drift along with the mean flow (and mean velocity) $\overline{U}$ to make sure only the turbulent fluctuations are measured. Evidently, such a set of devices, which is large enough in number and is fixed in a Lagrangean coordinate system moving with the mean flow is virtually impossible to realize in practice; and measurements have to be made on the basis of an Eulerian system with the measuring devices fixed w.r.t. the ground. This Eulerian frame can now be linked to the desired Lagrangean frame on the basis of a hypothesis, originally due to G.I. Taylor, that the sequence of turbulent fluctuations at a fixed point is statistically the same as if the spatial pattern of velocities were suddenly frozen and swept past the measuring device with the speed $\overline{U}$ of the mean flow. The set of several measuring devices fixed within a Lagrangean frame as described above can now be replaced by only one device in the Eulerian frame which measures in equi-spaced time intervals t which relate to the equi-spaced distances d of the Lagrangean frame in the following obvious way:

$$d = \overline{U} t \,, \tag{E50}$$

which is the mathematical formulation of the Taylor hypothesis.

Clearly, the Taylor hypothesis is important for the present work, where only a rather limited number of points (i.e., four) is monitored over a time period in an Eulerian frame fixed in fluid space. Therefore, a Fourier analysis can only be done w.r.t. time, which gives a frequency spectrum. However, as is evident from (E50), the Taylor hypothesis holds in the context of wavespace and frequency as well:

$$k = f/\overline{U} \tag{E51}$$

with k and f as wave number and frequency respectively.

APPENDIX F

1. TRAJECTORY PROJECTIONS AND SPECTRAL GRAPHS

◇ denotes zero on trajectory scales

○ denotes beginning of trajectory

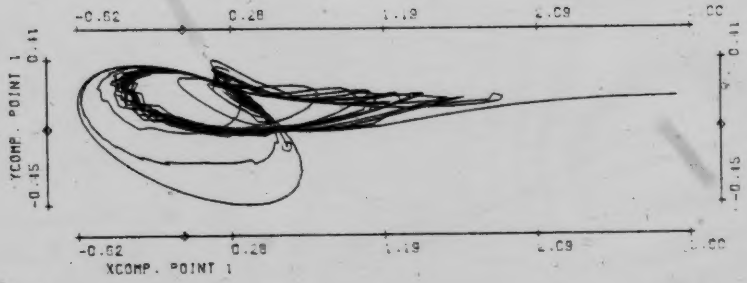
Z denotes end of trajectory

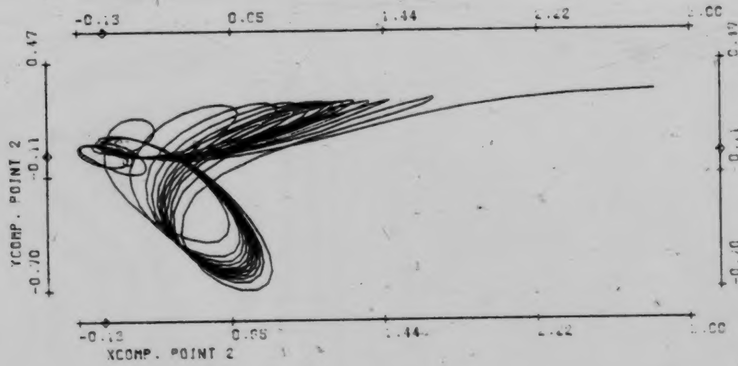
No spectral graphs S1, S2, S10, S11 are provided, since no spectral analysis was performed for the relating trajectories of Figures T1, T2, T10, T11. All trajectories belong to Flow IV except T1 and T2, which belong to Flow III.

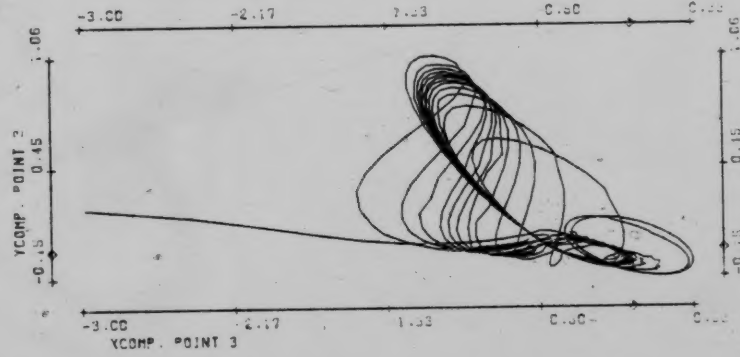
FIGURE T1

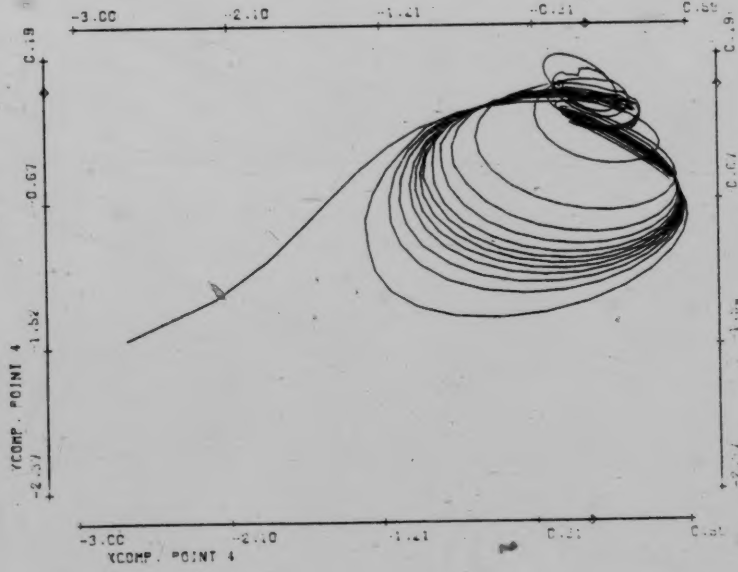
Bifurcation parameter η	.55
No. of timesteps	851
Stepsize	.03
Init. condition for $\vec{V}_2$	.001 (all)
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.66

T1(1)

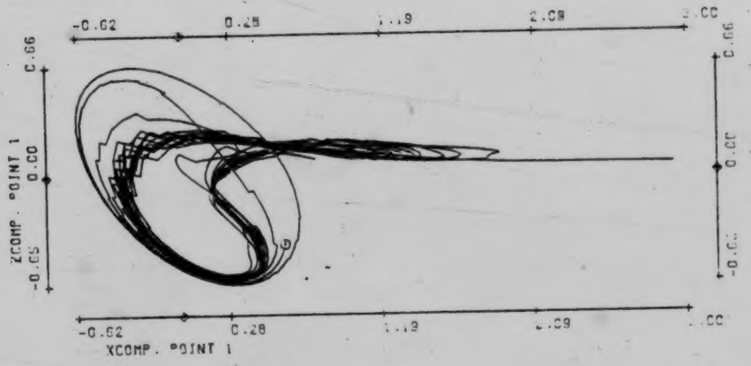

-0.62
0.28
1.19
2.09
3.00
YCOMP. POINT 1
0.41
-0.15
XCOMP. POINT 1

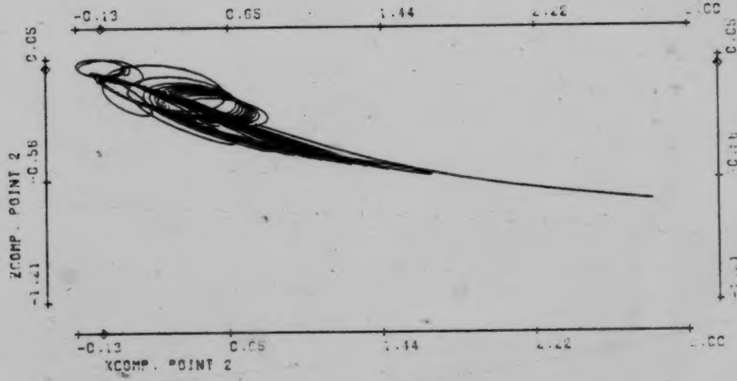

-0.13
0.65
1.44
2.22
3.00
YCOMP. POINT 2
0.47
-0.11
-0.70
XCOMP. POINT 2


-3.00
-2.17
-1.33
-0.50
YCOMP. POINT 3
1.06
0.45
-0.15
XCOMP. POINT 3

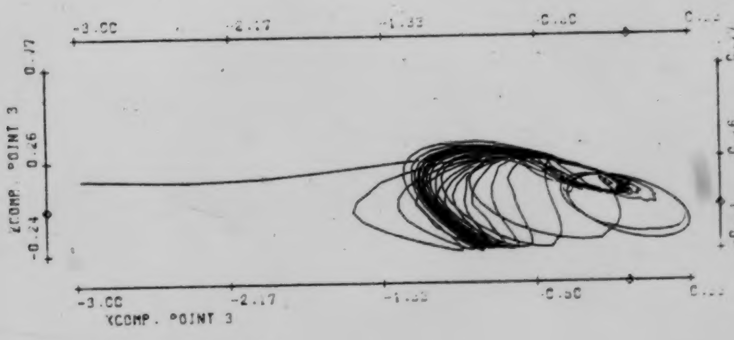

-3.00
-2.10
-1.21
-0.31
0.59
YCOMP. POINT 4
0.19
-0.67
-1.52
-2.37
XCOMP. POINT 4

T1(2)

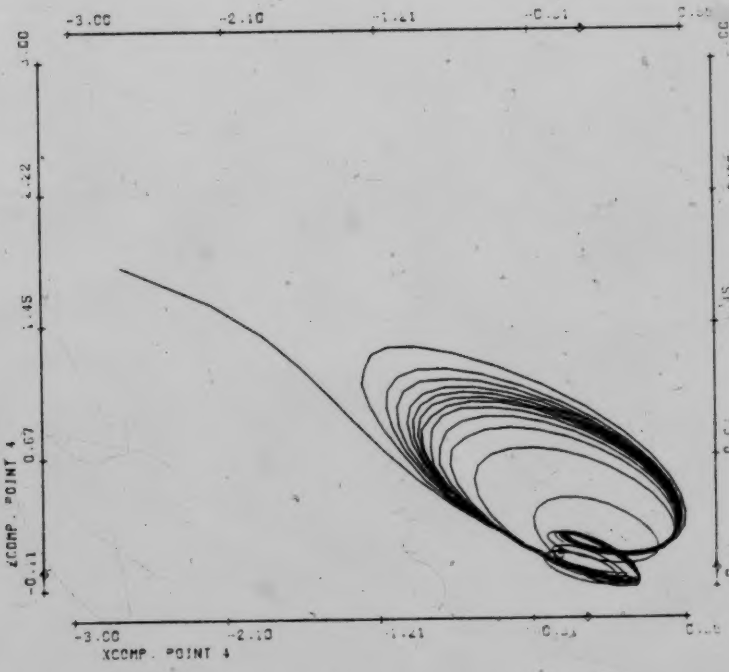

-0.62
0.25
1.19
2.09
3.00
0.66
0.00
-0.65
ZCOMP. POINT 1
-0.62
0.28
1.19
2.09
3.00
XCOMP. POINT 1


-0.13
0.65
1.44
2.22
3.00
0.05
-0.56
-1.21
ZCOMP. POINT 2
-0.13
0.65
1.44
2.22
3.00
XCOMP. POINT 2

T1(3)

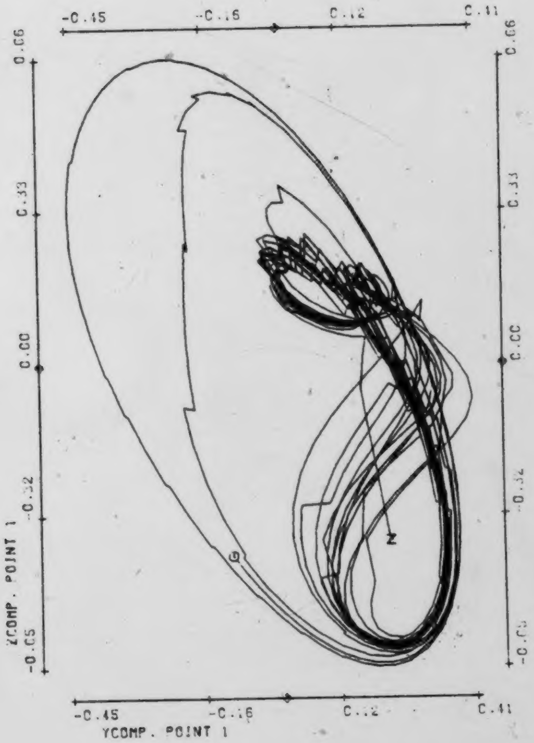


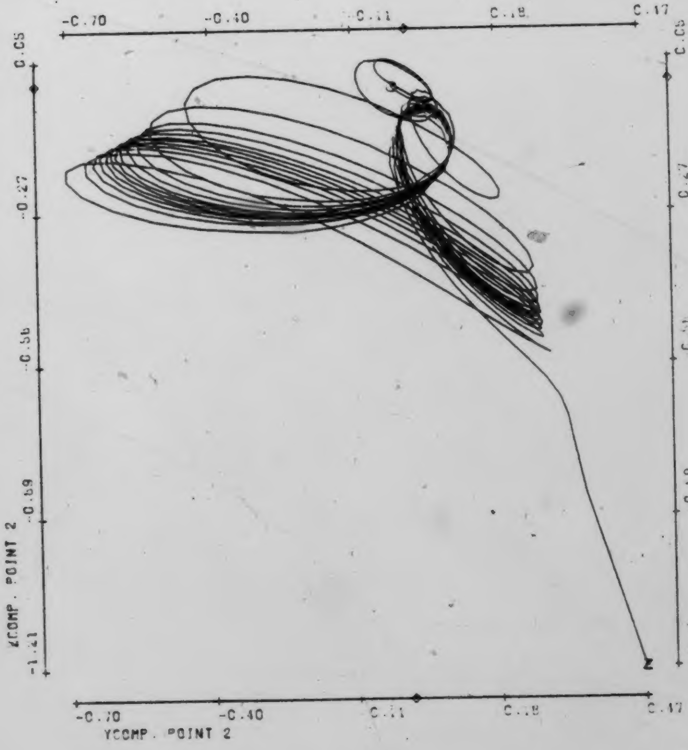
-3.00
-2.17
-1.33
-0.50
-0.24
0.26
0.77
ZCOMP. POINT 3
XCOMP. POINT 3



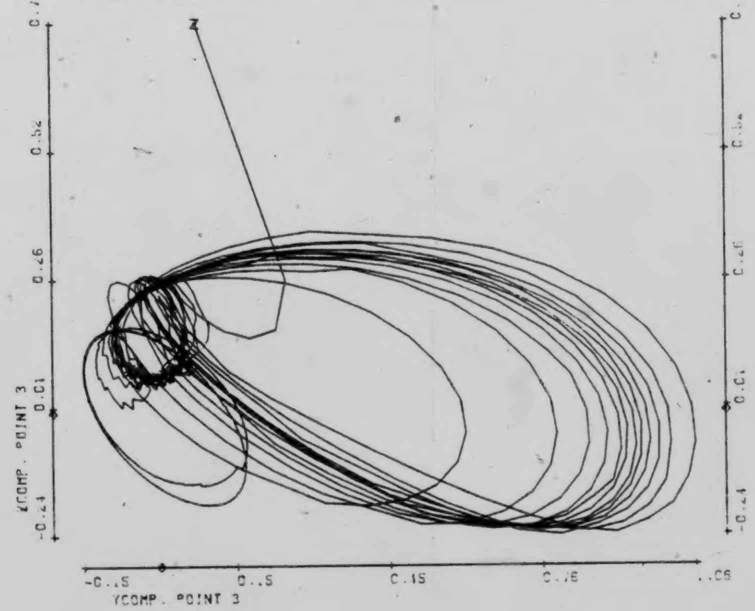
-3.00
-2.10
-1.21
-0.31
-0.11
0.67
1.45
2.22
3.00
ZCOMP. POINT 4
XCOMP. POINT 4

T1(4)


-0.45
-0.16
0.12
0.41
0.66
0.33
0.00
-0.32
-0.65
Z
G
ZCOMP. POINT 1
YCOMP. POINT 1


-0.70
-0.40
-0.11
0.18
0.47
0.05
-0.27
-0.58
-0.89
-1.21
Z
ZCOMP. POINT 2
YCOMP. POINT 2

T1(5)


-0..5
0..5
0.15
0.76
1.06
0.77
0.52
0.26
0.01
-0.24
Z
ZCOMP. POINT 3
YCOMP. POINT 3

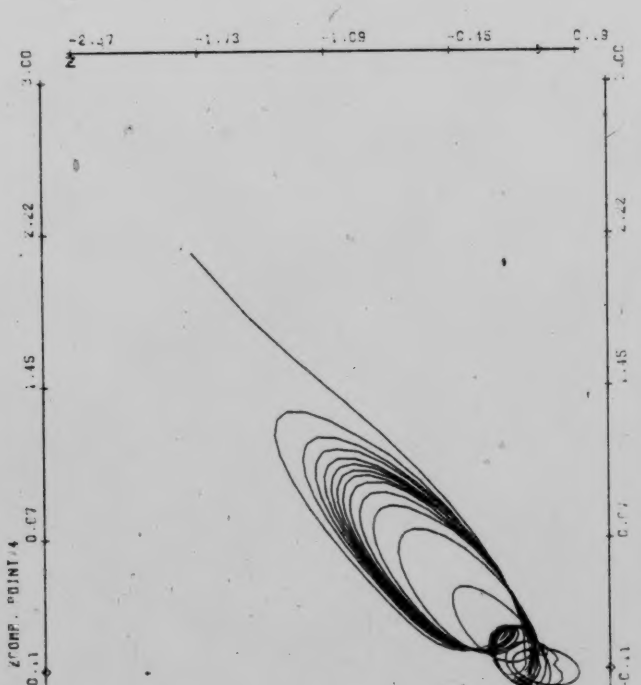
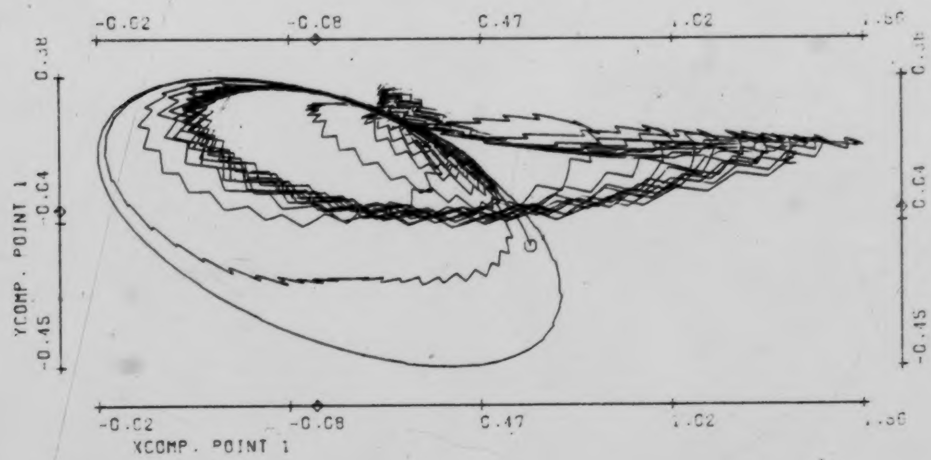
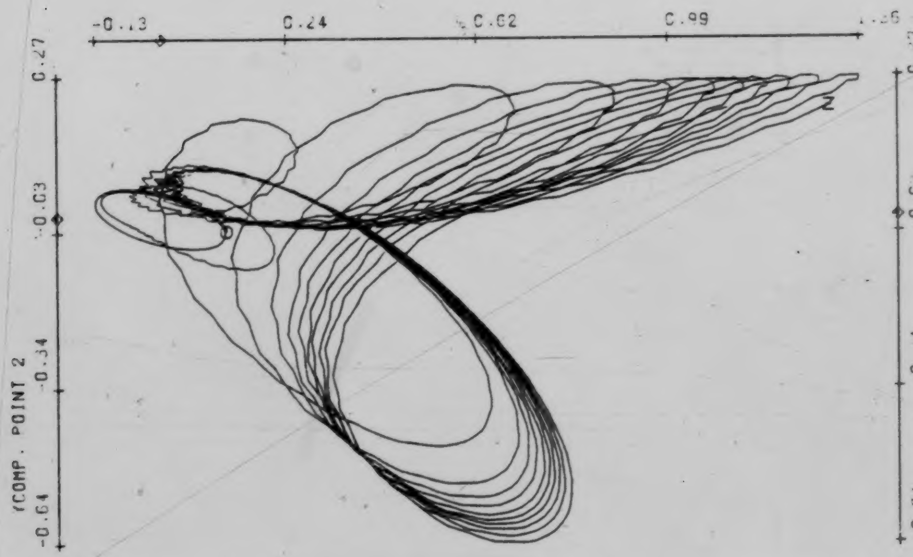

-2.37
-1.73
-1.09
-0.45
0.9
3.00
2.22
1.45
0.67
-0.11
Z
ZCOMP. POINT 4
YCOMP. POINT 4

FIGURE T2

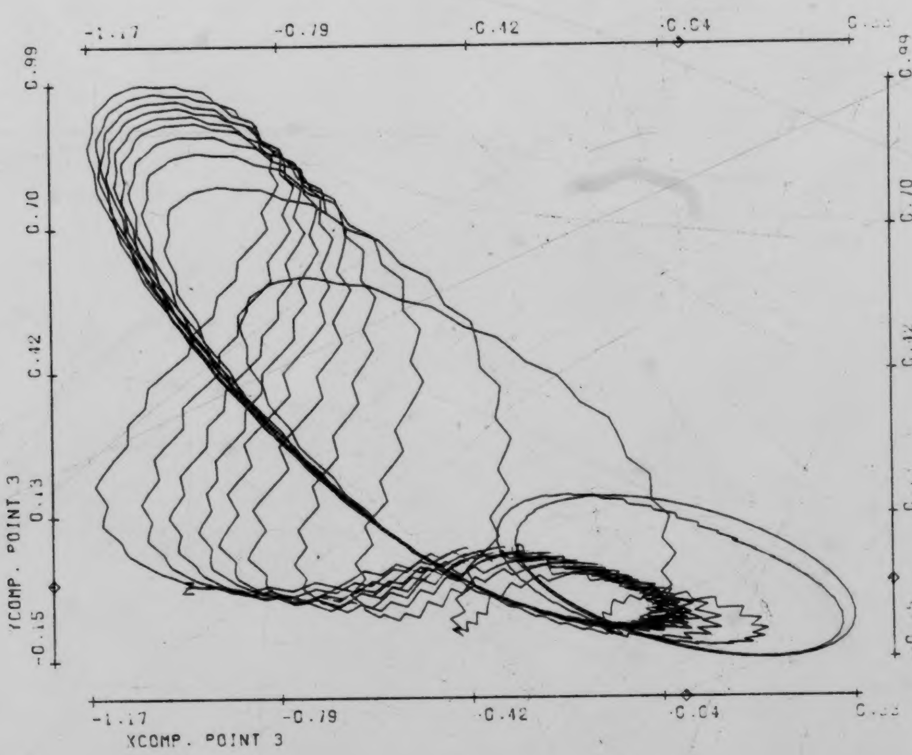
Bifurcation parameter η	.55
No. of timesteps	2000
Stepsize	.01
Init. condition for $\vec{V}_2$	.001 (all)
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.66

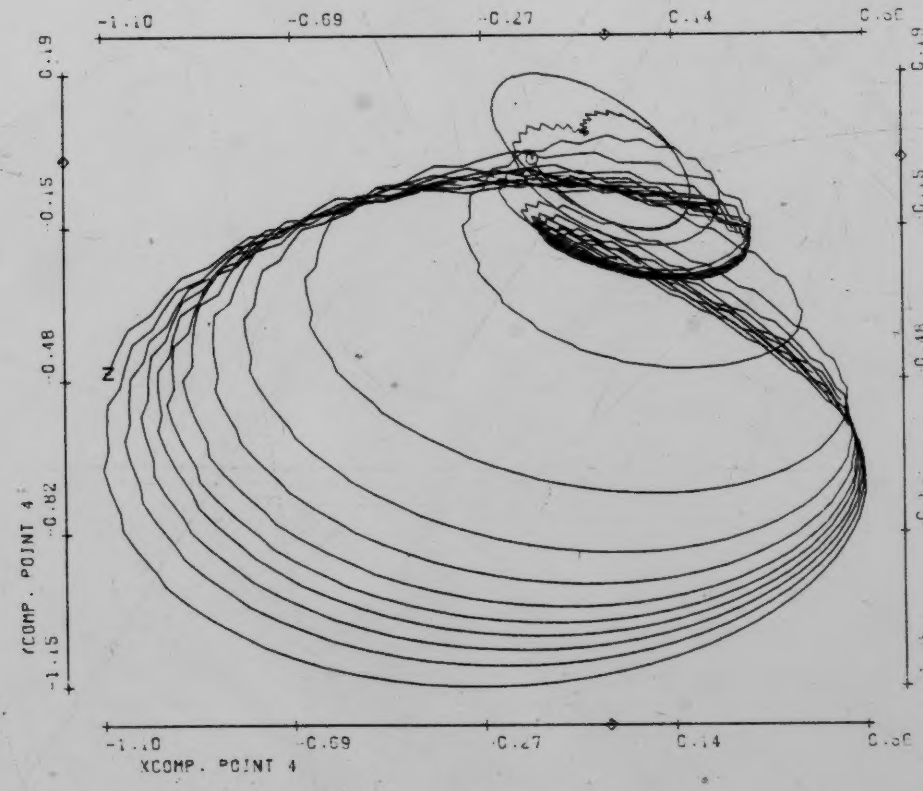
T2(1)


-0.62
-0.08
0.47
1.02
1.56
0.38
-0.04
-0.45
YCOMP. POINT 1
XCOMP. POINT 1
G

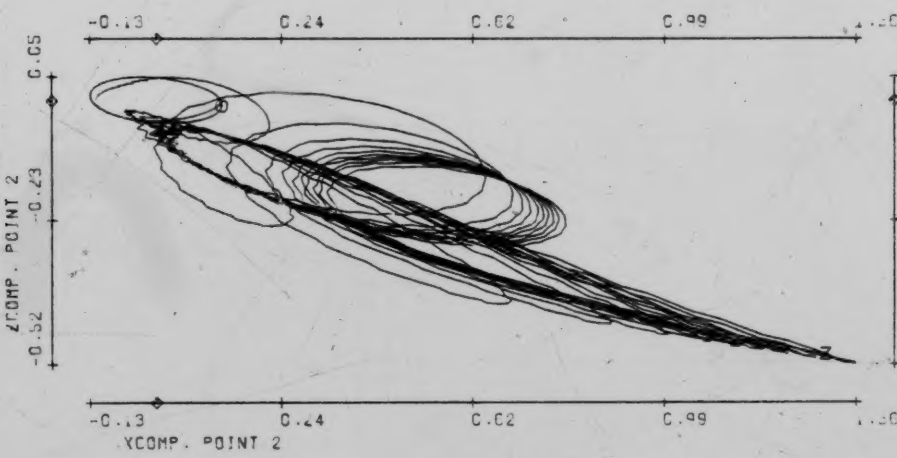

-0.13
0.24
0.62
0.99
1.36
0.27
-0.03
-0.34
-0.64
YCOMP. POINT 2
XCOMP. POINT 2
O
Z

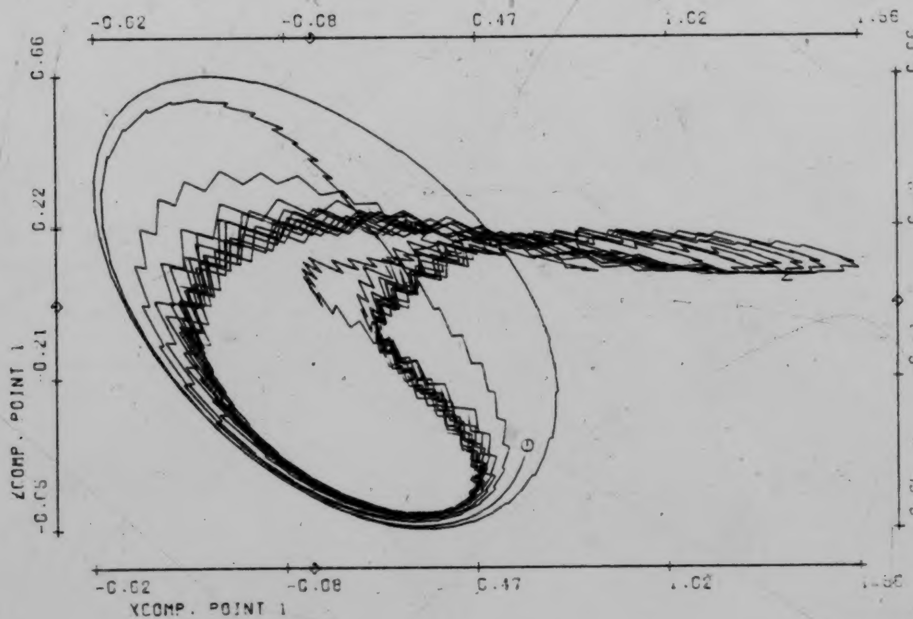
T2(2)


-1.17
-0.79
-0.42
-0.04
0.99
0.70
0.42
0.13
-0.15
Z
YCOMP. POINT 3
XCOMP. POINT 3

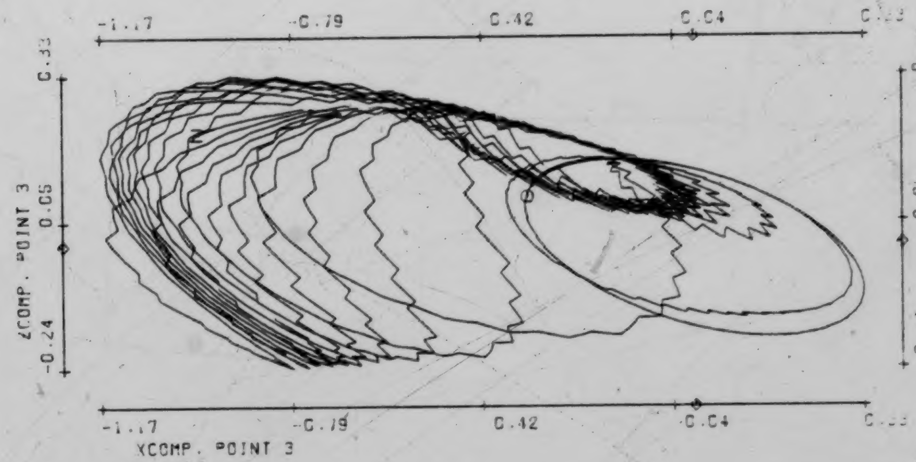

-1.10
-0.69
-0.27
0.14
0.56
0.19
-0.15
-0.48
-0.82
-1.15
Z
YCOMP. POINT 4
XCOMP. POINT 4

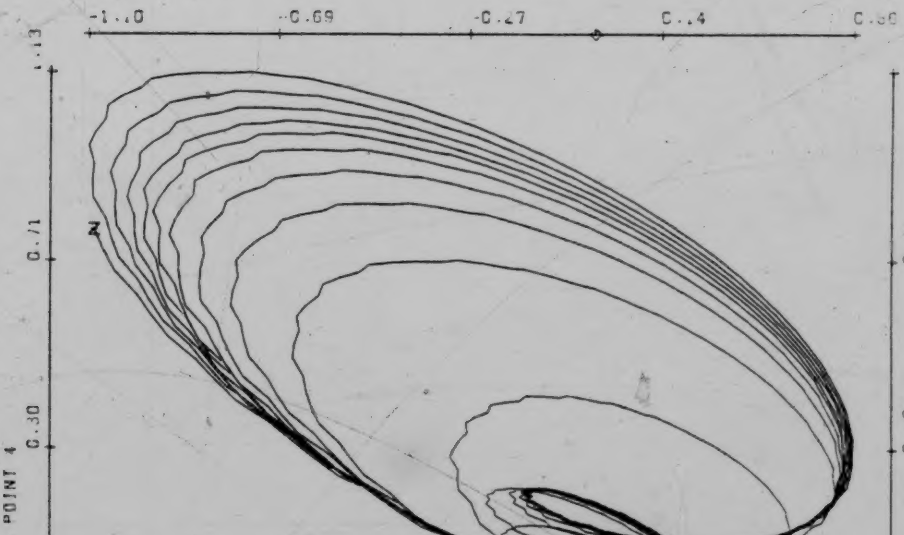
T2(3)

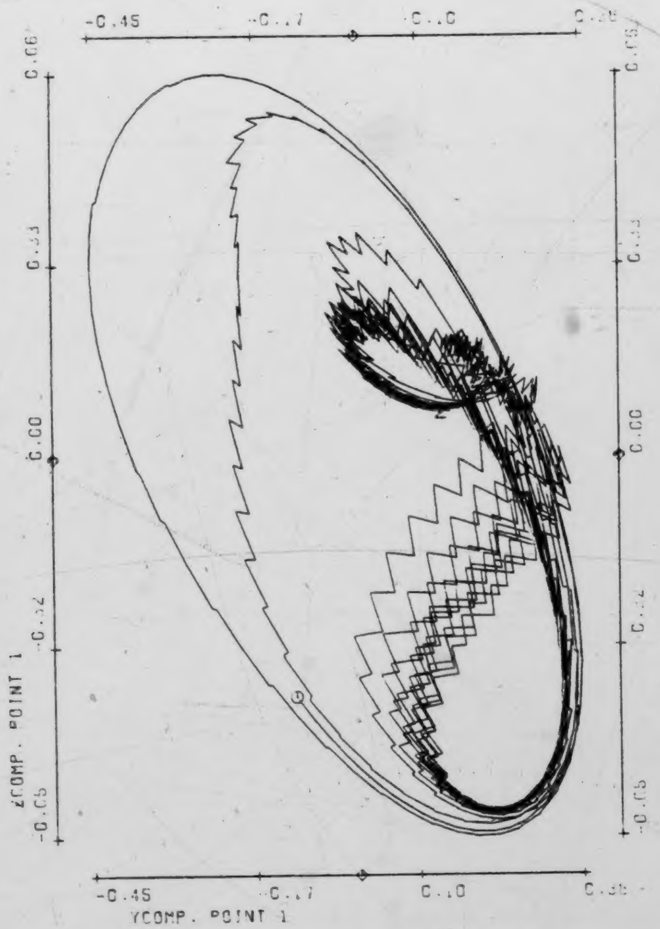

-0.13
0.24
0.62
0.99
1.36
0.05
-0.23
-0.52
ZCOMP. POINT 2
XCOMP. POINT 2

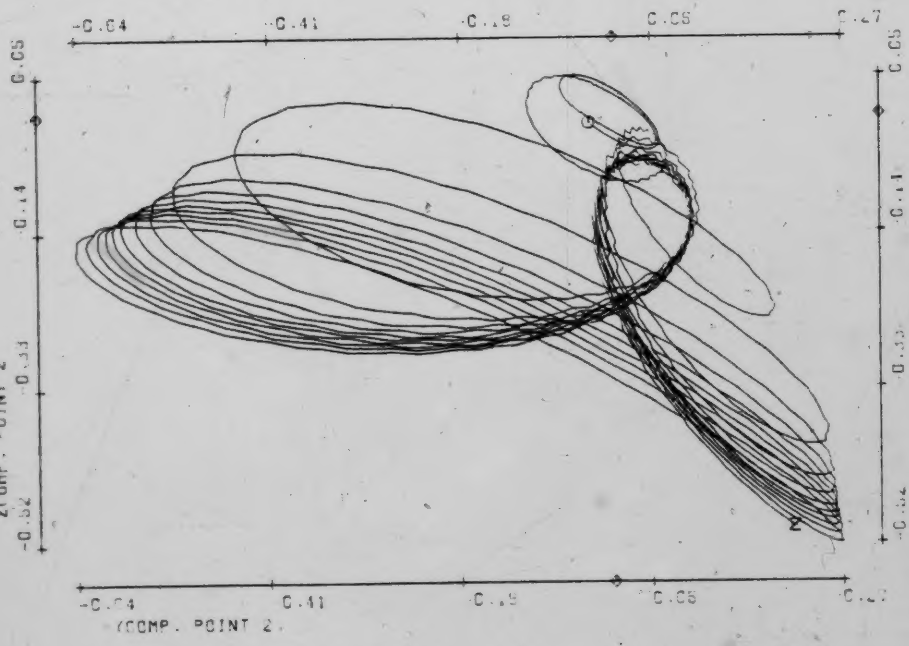

-0.62
-0.08
0.47
1.02
1.56
0.66
0.22
-0.21
-0.65
ZCOMP. POINT 1
XCOMP. POINT 1

T2(4)

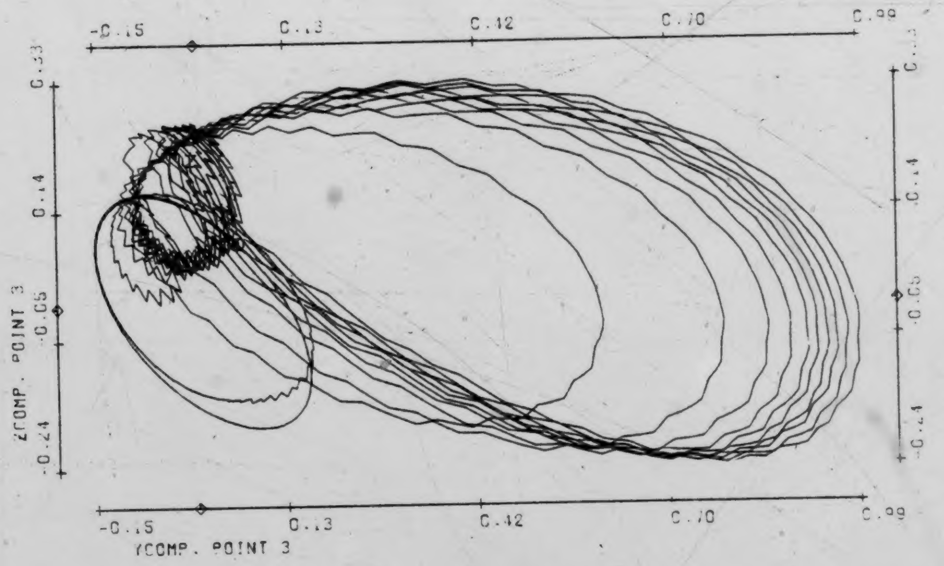

-1.17
-0.79
0.42
0.04
0.33
0.05
-0.24
ZCOMP. POINT 3
XCOMP. POINT 3


-1.10
-0.69
-0.27
0.14
0.56
1.13
0.71
0.30
-0.11
ZCOMP. POINT 4
XCOMP. POINT 4


-0.45
-0.17
0.10
0.38
0.06
0.03
0.00
-0.02
-0.05
ZCOMP. POINT 1
YCOMP. POINT 1


-0.64
-0.41
-0.18
0.05
0.27
0.05
-0.14
-0.33
-0.52
ZCOMP. POINT 2
YCOMP. POINT 2

T2(6)


-0.15
0.13
0.42
0.70
0.99
0.33
0.14
-0.05
-0.24
ZCOMP. POINT 3
YCOMP. POINT 3

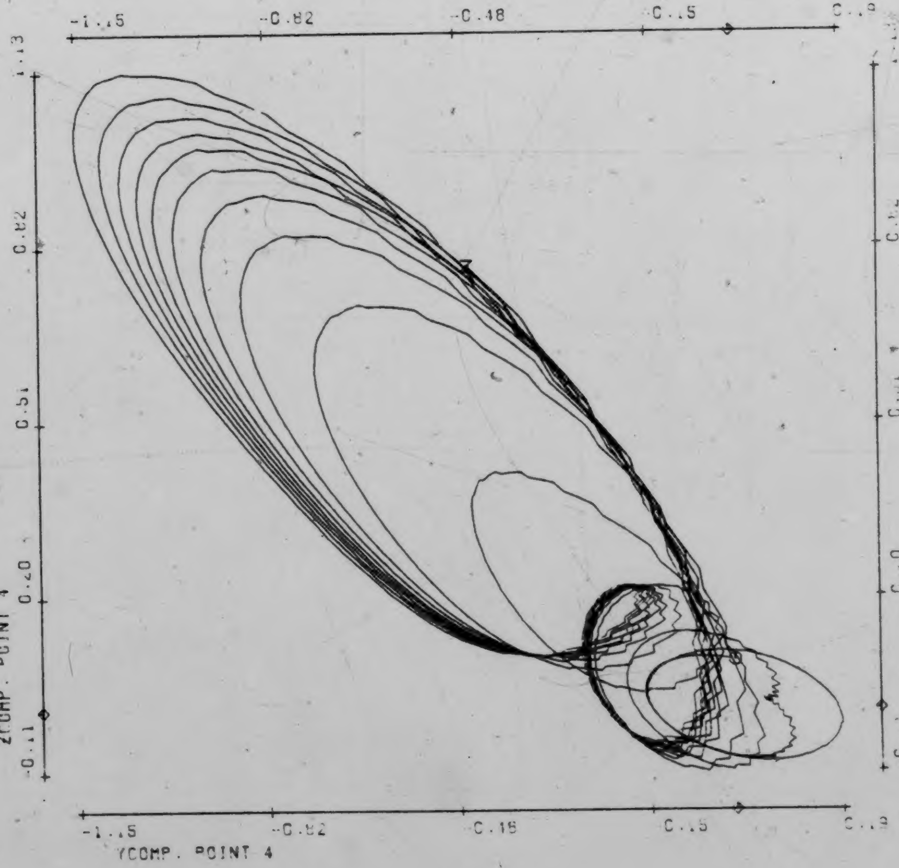
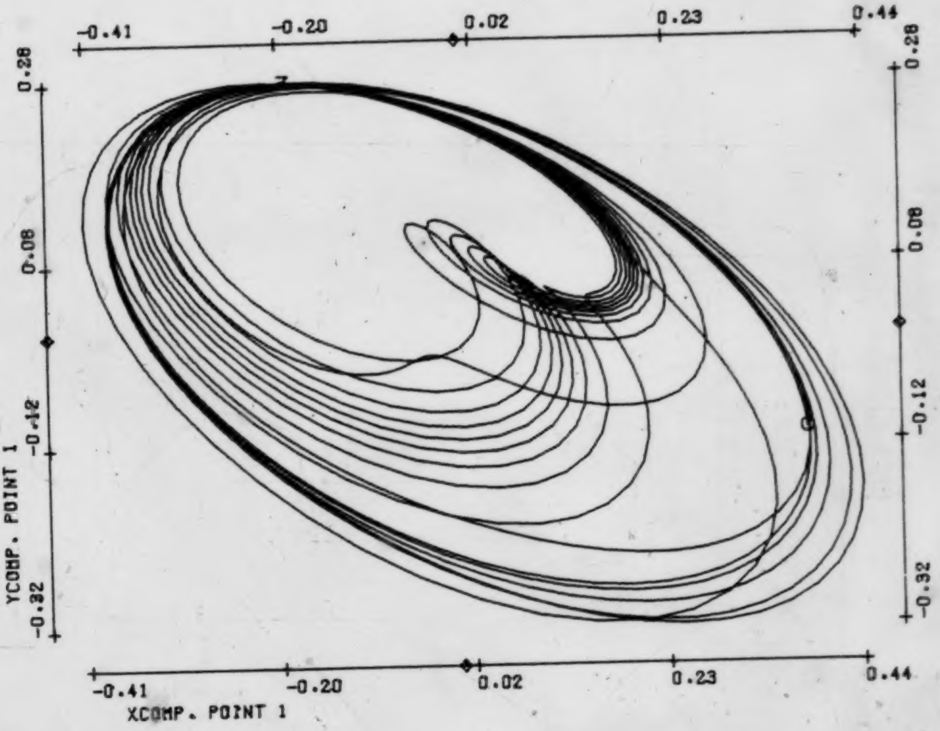
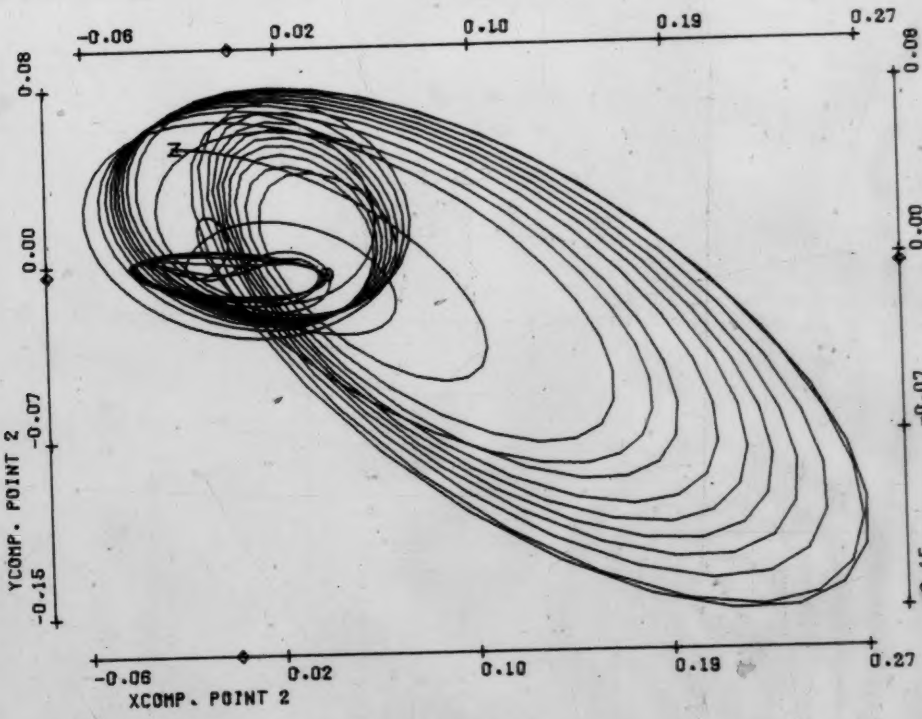

-1.15
-0.82
-0.48
-0.15
0.19
1.13
0.82
0.51
0.20
-0.11
ZCOMP. POINT 4
YCOMP. POINT 4

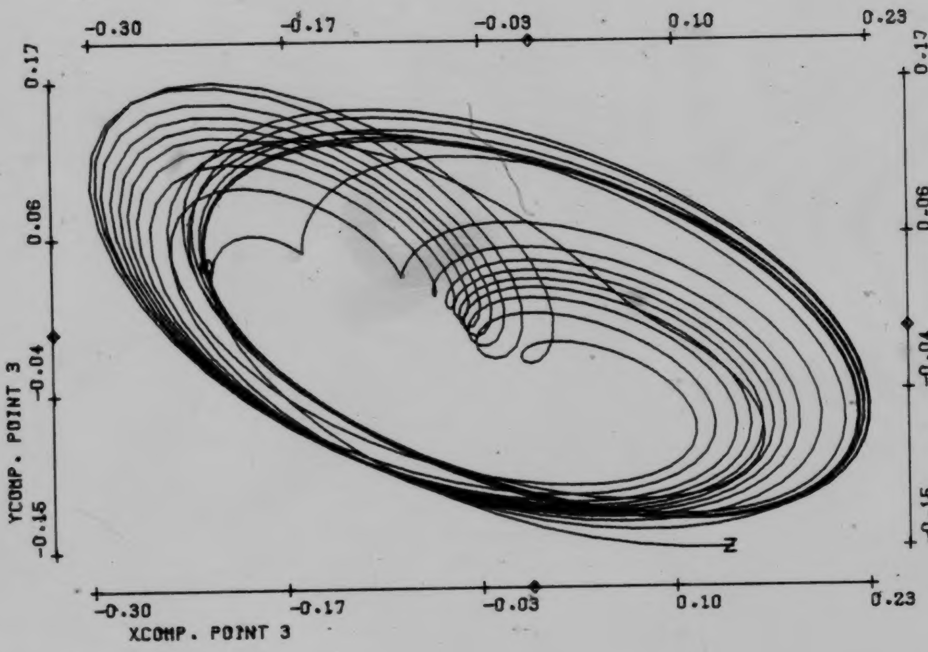
FIGURE T3

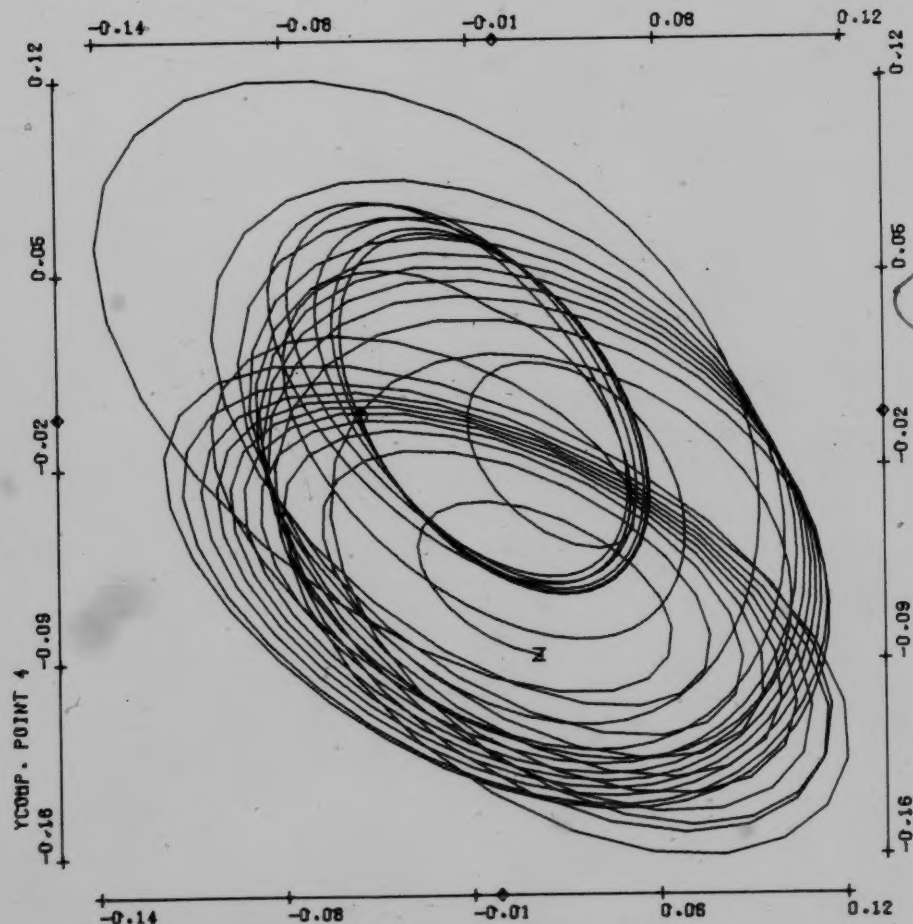
Bifurcation parameter η	.3
No. of timesteps	1000
Stepsize	.03
Init. condition for $\vec{V}_2$	.001 (all)
Average size of spatial random perturbations	n/a
Average no. of timestepts between random perturbations	n/a
Period of $\vec{V}_1$	1.68
Mean Velocity:	
Point 1	2.20
2	2.03
3	2.95
4	1.85

T3(1)

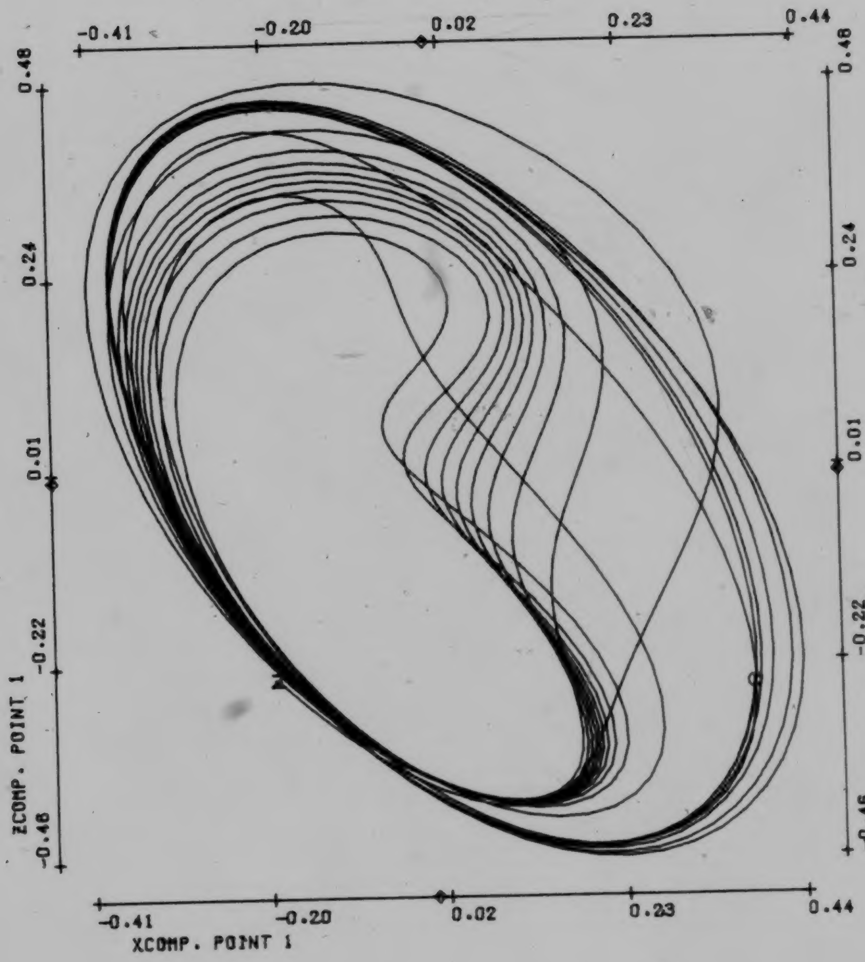

-0.41
-0.20
0.02
0.23
0.44
0.28
0.08
-0.12
-0.32
YCOMP. POINT 1
XCOMP. POINT 1

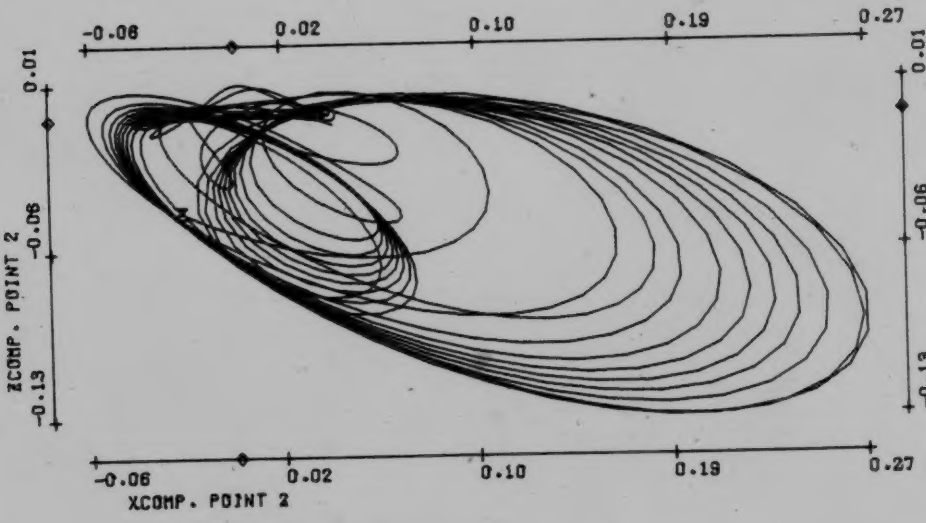

-0.06
0.02
0.10
0.19
0.27
0.08
0.00
-0.07
-0.15
YCOMP. POINT 2
XCOMP. POINT 2


-0.30
-0.17
-0.03
0.10
0.23
0.17
0.06
-0.04
-0.15
YCOMP. POINT 3
XCOMP. POINT 3

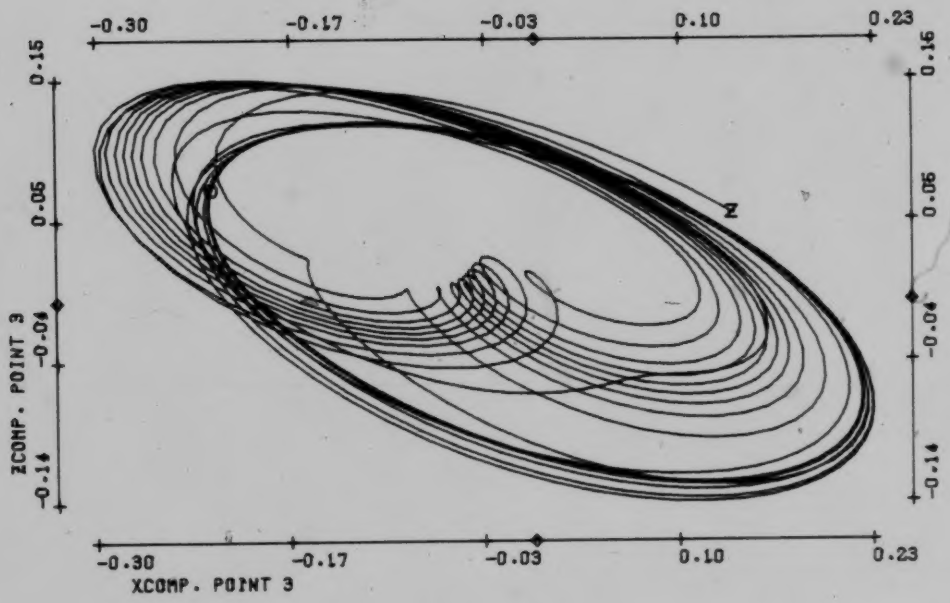

-0.14
-0.08
-0.01
0.06
0.12
0.12
0.05
-0.02
-0.09
-0.16
YCOMP. POINT 4
XCOMP. POINT 4

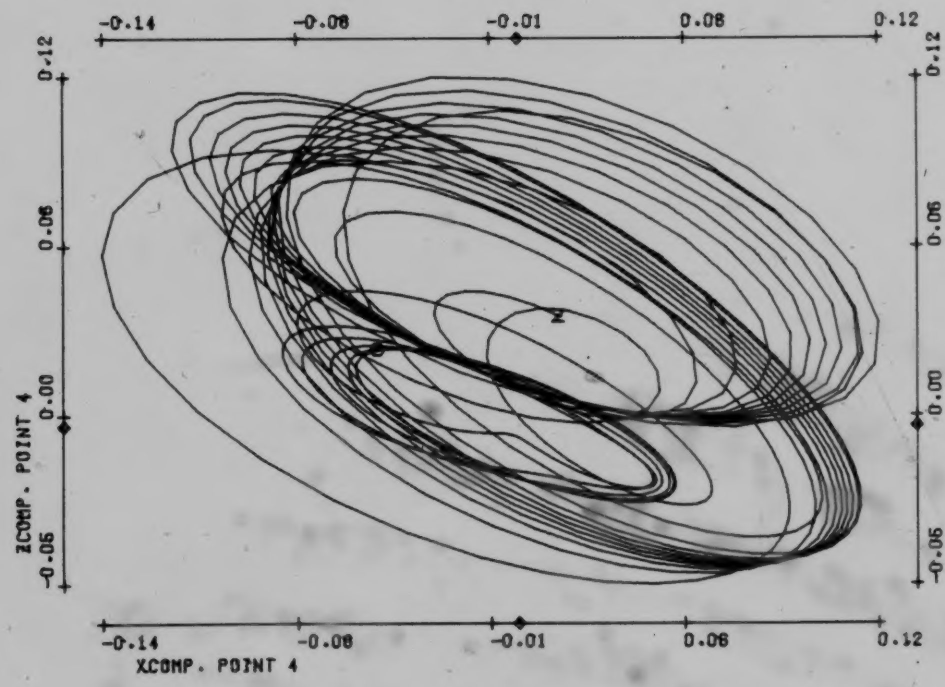
T3(3)


ZCOMP. POINT 1
XCOMP. POINT 1
-0.41
-0.20
0.02
0.23
0.44
0.48
0.24
0.01
-0.22
-0.46

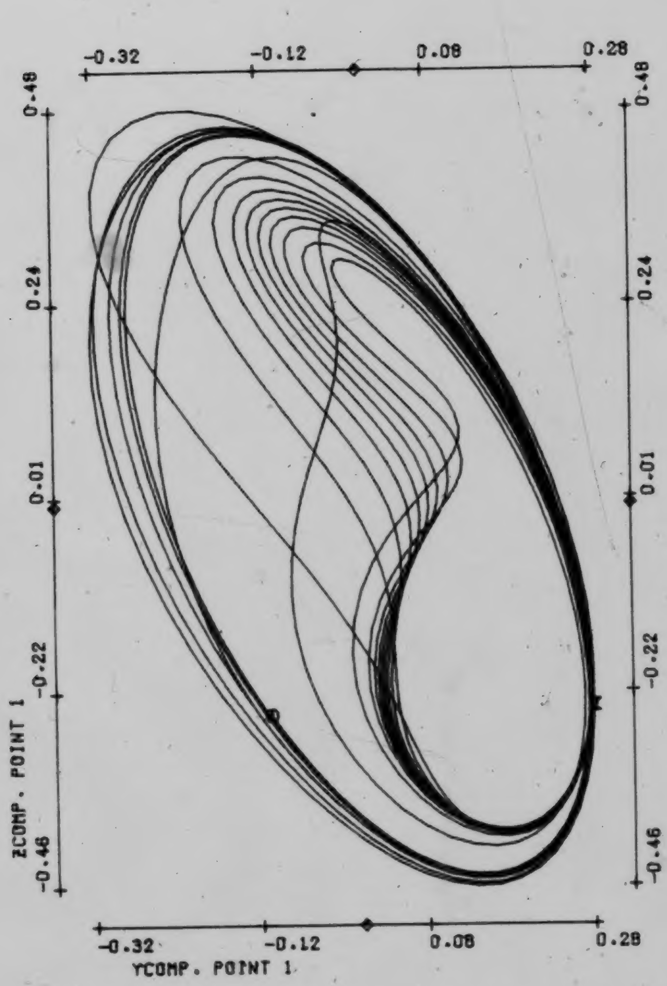

ZCOMP. POINT 2
XCOMP. POINT 2
-0.06
0.02
0.10
0.19
0.27
0.01
-0.06
-0.13

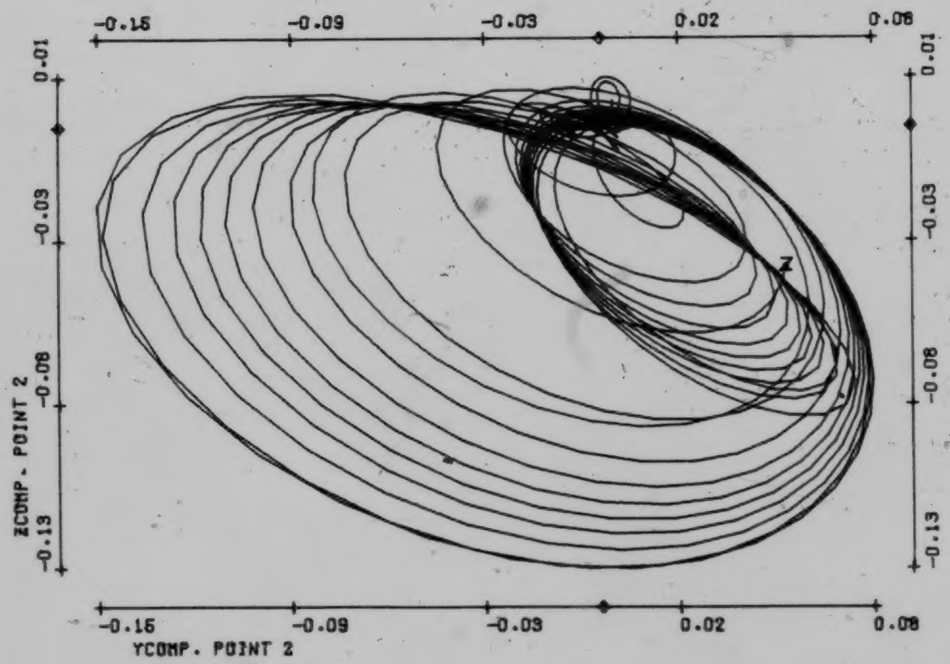
T3(4)

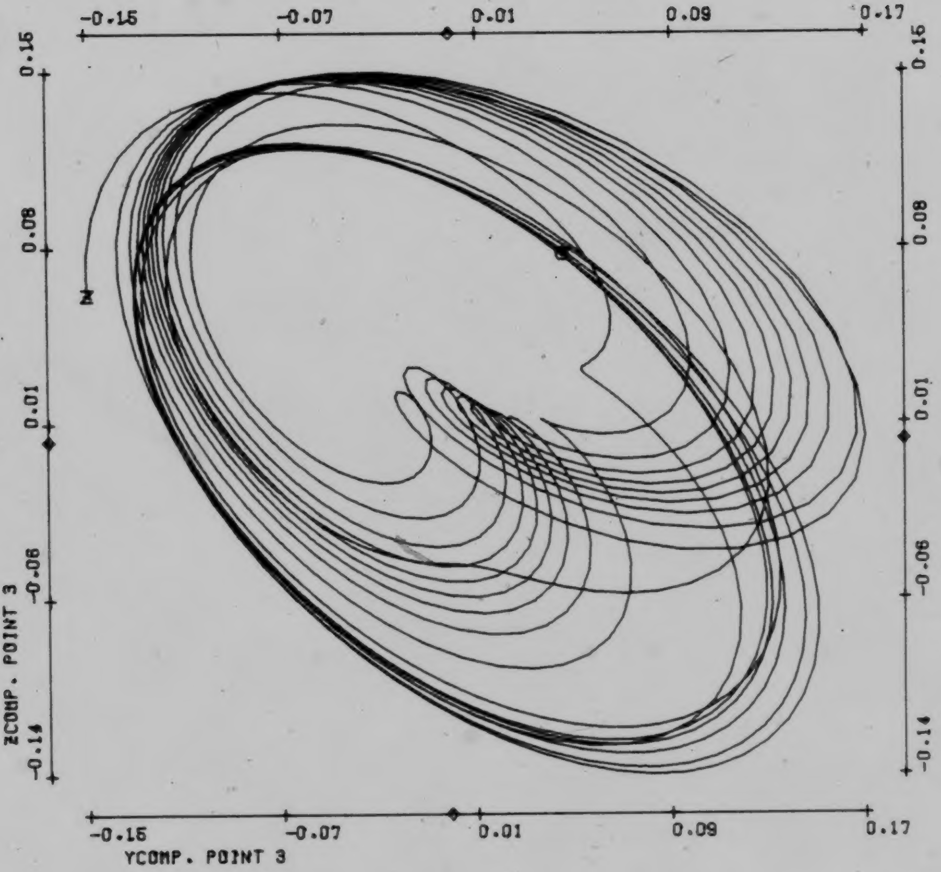

-0.30
-0.17
-0.03
0.10
0.23
0.15
0.05
-0.04
-0.14
0.16
0.06
-0.04
-0.14
Z
O
ZCOMP. POINT 3
XCOMP. POINT 3


-0.14
-0.08
-0.01
0.06
0.12
0.12
0.06
0.00
-0.06
0.12
0.06
0.00
-0.06
Z
O
ZCOMP. POINT 4
XCOMP. POINT 4

T3(5)


-0.32
-0.12
0.08
0.28
0.48
0.24
0.01
-0.22
-0.46
ZCOMP. POINT 1
YCOMP. POINT 1


-0.15
-0.09
-0.03
0.02
0.08
0.01
-0.03
-0.08
-0.13
ZCOMP. POINT 2
YCOMP. POINT 2


-0.15
-0.07
0.01
0.09
0.17
0.15
0.08
0.01
-0.06
-0.14
ZCOMP. POINT 3
YCOMP. POINT 3
Z

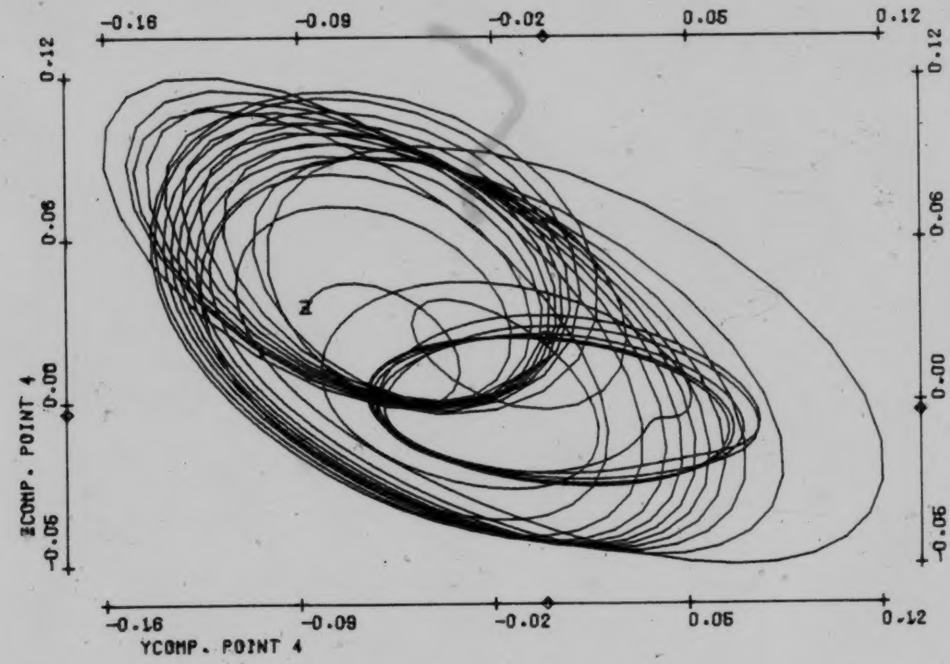
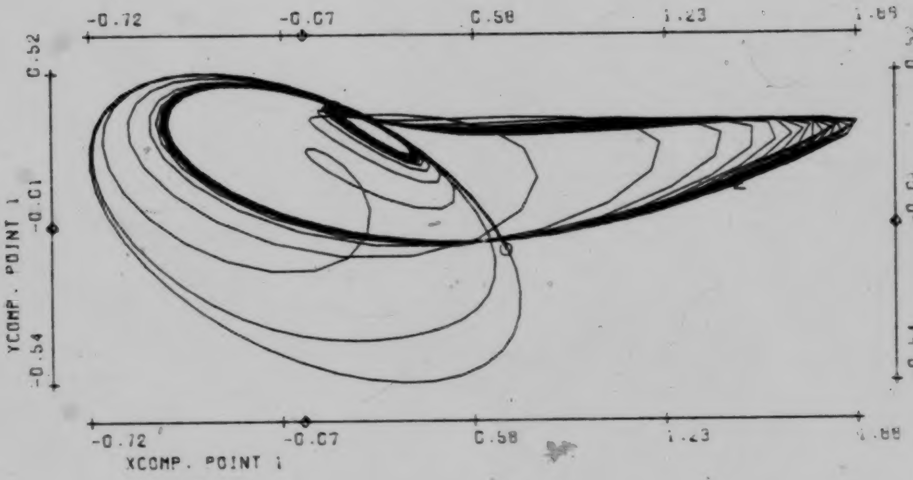
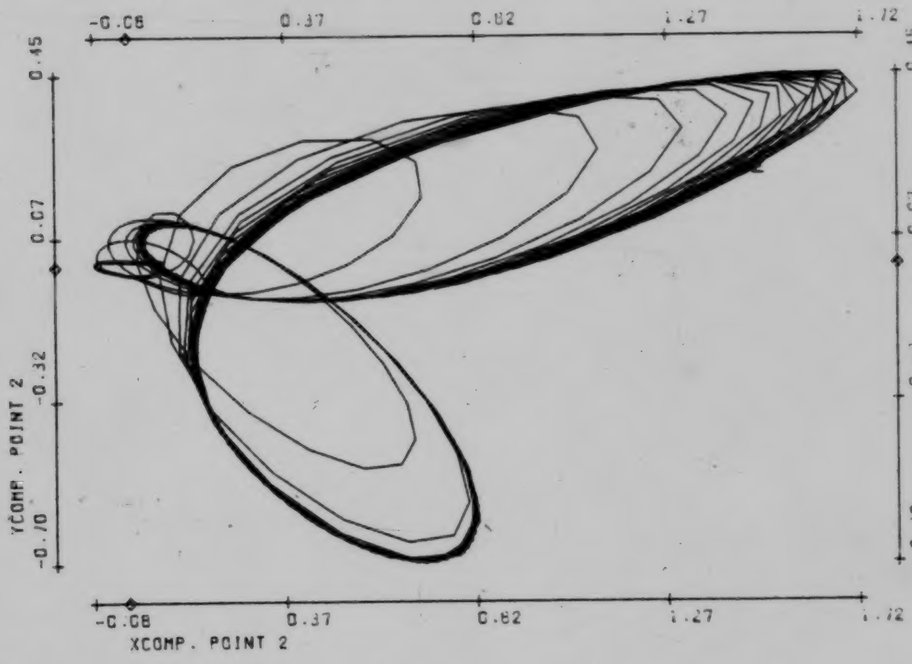

-0.16
-0.09
-0.02
0.05
0.12
0.12
0.06
0.00
-0.06
ZCOMP. POINT 4
YCOMP. POINT 4
Z

FIGURE T4

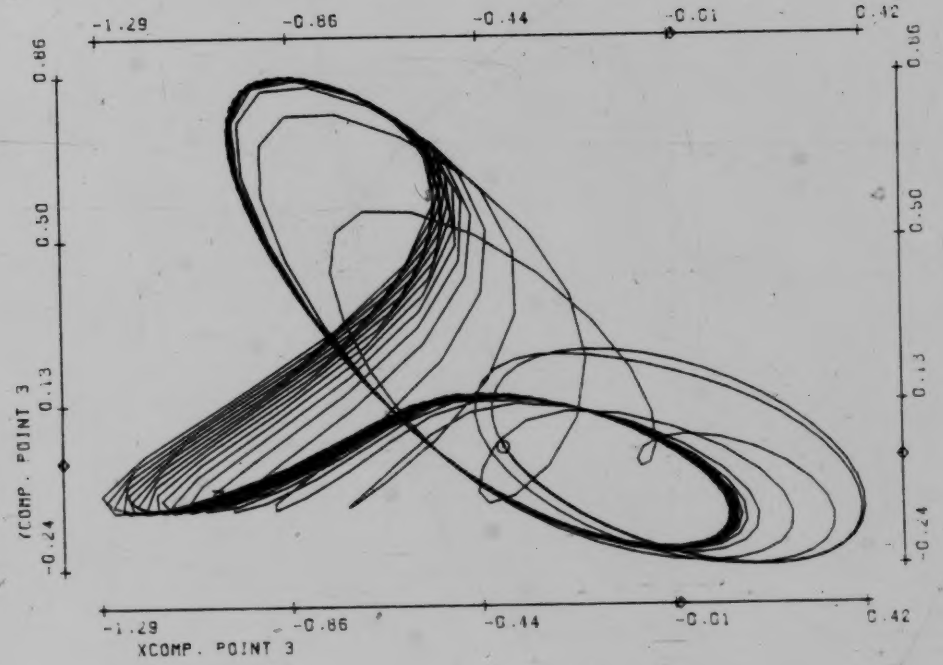
Bifurcation parameter η	.449
No. of timesteps	800
Stepsize	.04
Init. condition for $\vec{V}_2$	.001(X-comp.Pt.1) rest = 0
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.55
Mean velocity:	
Point 1	1.38
2	1.76
3	2.84
4	1.95

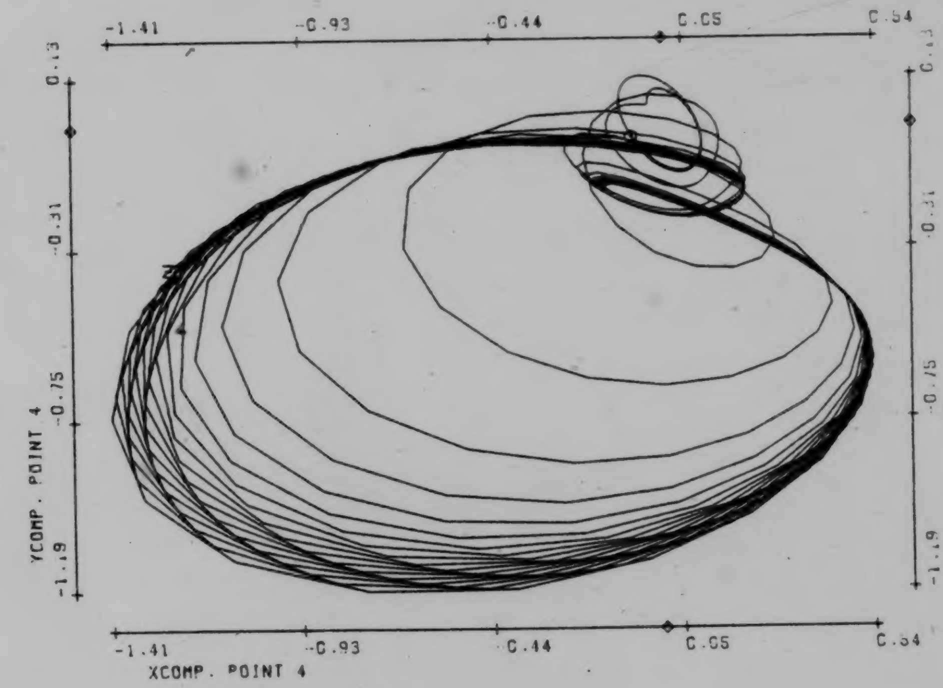
T4(1)


-0.72
-0.07
0.58
1.23
1.88
0.52
-0.01
-0.54
YCOMP. POINT 1
XCOMP. POINT 1

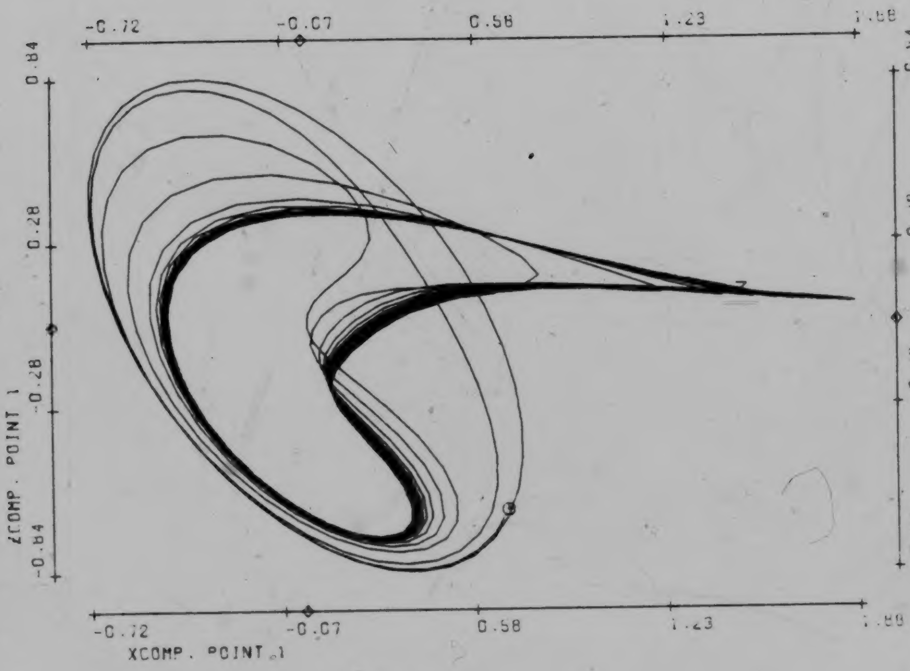

-0.08
0.37
0.82
1.27
1.72
0.45
0.07
-0.32
-0.70
YCOMP. POINT 2
XCOMP. POINT 2

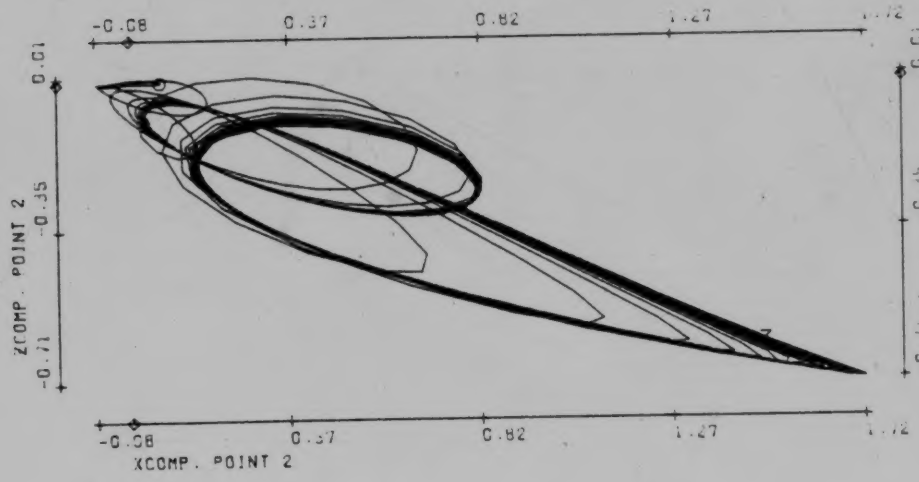
T4(2)


-1.29
-0.86
-0.44
-0.01
0.42
0.86
0.50
0.13
-0.24
YCOMP. POINT 3
XCOMP. POINT 3

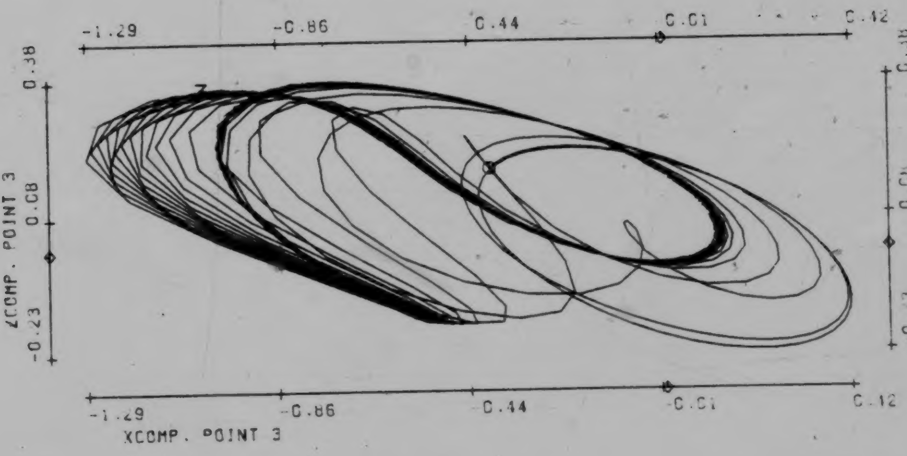

-1.41
-0.93
-0.44
0.05
0.54
0.13
-0.31
-0.75
-1.19
YCOMP. POINT 4
XCOMP. POINT 4

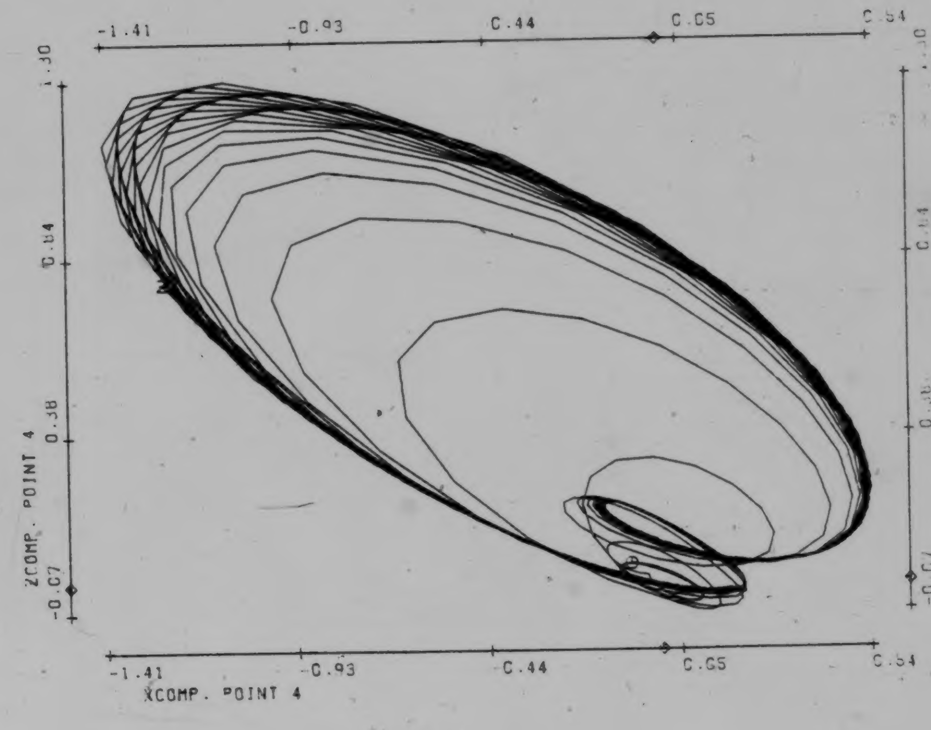
T4(3)


-0.72
-0.07
0.58
1.23
1.88
0.84
0.28
-0.28
-0.84
ZCOMP. POINT 1
XCOMP. POINT 1

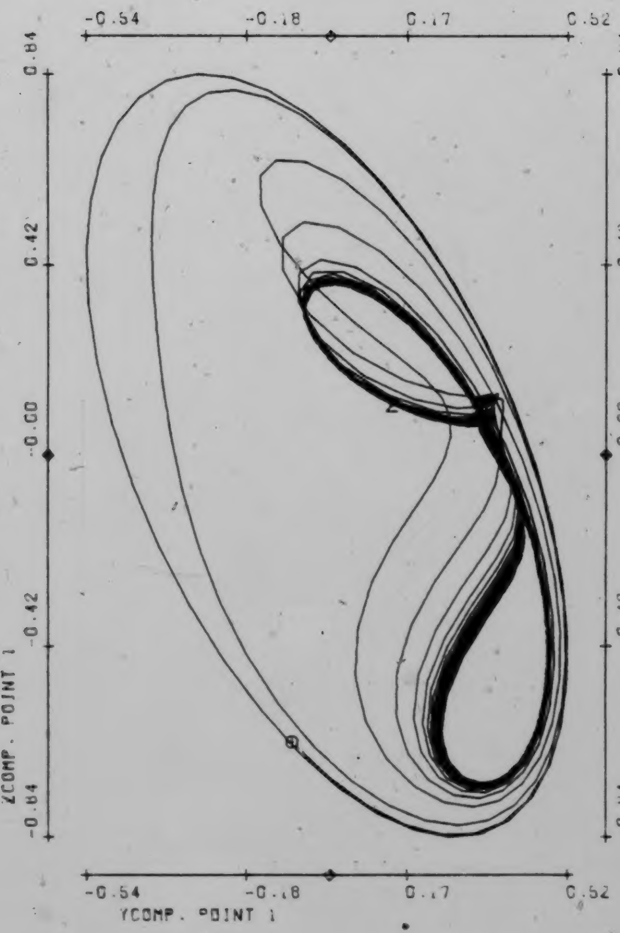

-0.08
0.37
0.82
1.27
1.72
0.01
-0.35
-0.71
ZCOMP. POINT 2
XCOMP. POINT 2

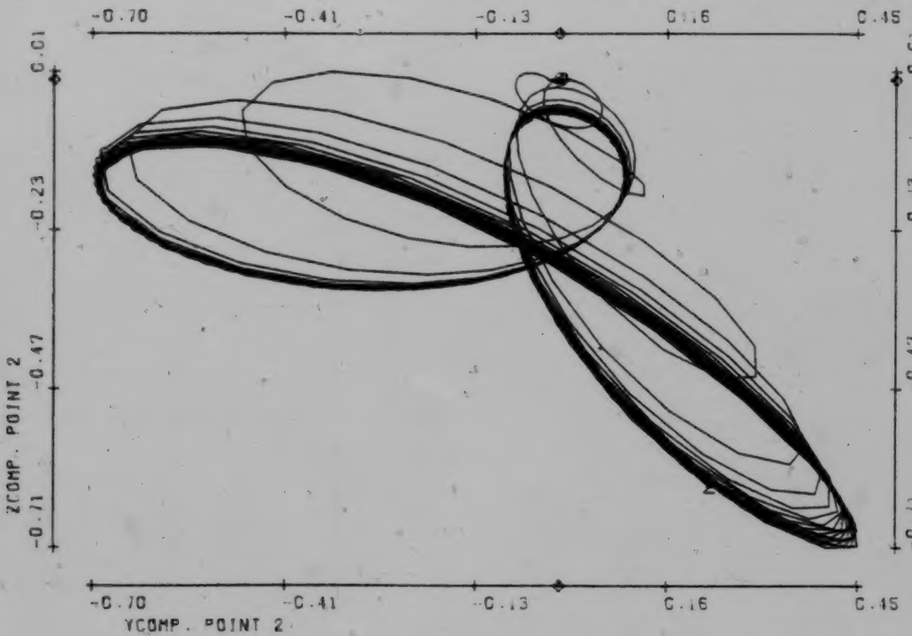
T4(4)


-1.29
-0.86
0.44
0.01
0.42
0.38
0.08
-0.23
ZCOMP. POINT 3
XCOMP. POINT 3

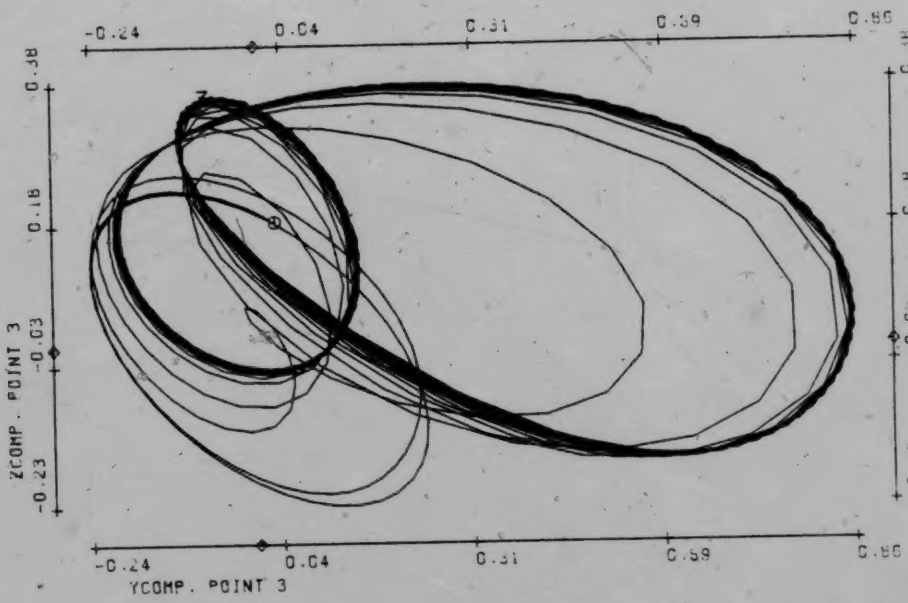

-1.41
-0.93
0.44
0.05
0.54
1.30
0.84
0.38
-0.07
ZCOMP. POINT 4
XCOMP. POINT 4

T4(5)


-0.54
-0.18
0.17
0.52
0.84
0.42
-0.00
-0.42
-0.84
ZCOMP. POINT 1
YCOMP. POINT 1


-0.70
-0.41
-0.13
0.16
0.45
0.01
-0.23
-0.47
-0.71
ZCOMP. POINT 2
YCOMP. POINT 2

T4(6)


-0.24
0.04
0.31
0.59
0.86
0.38
0.18
-0.03
-0.23
ZCOMP. POINT 3
YCOMP. POINT 3

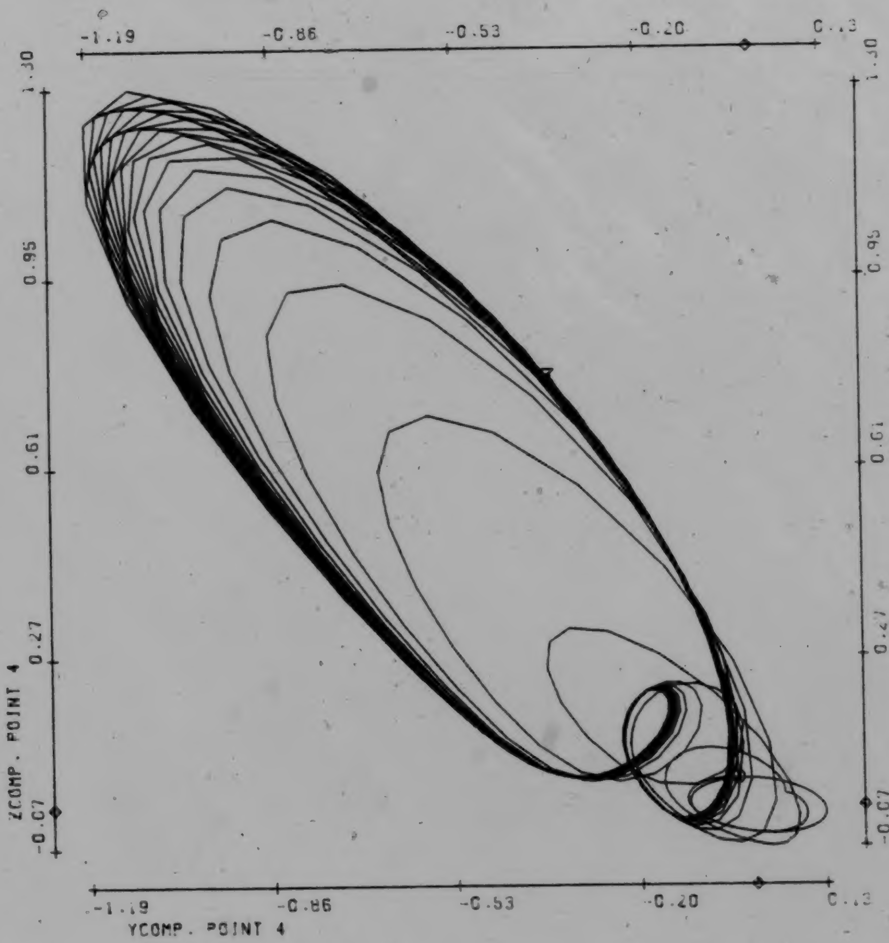
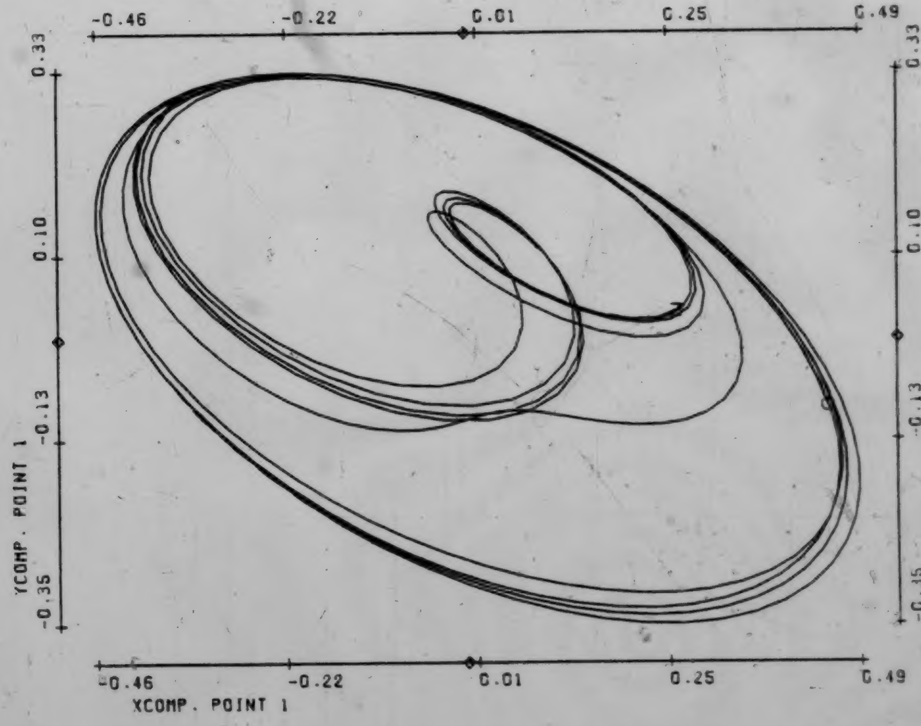
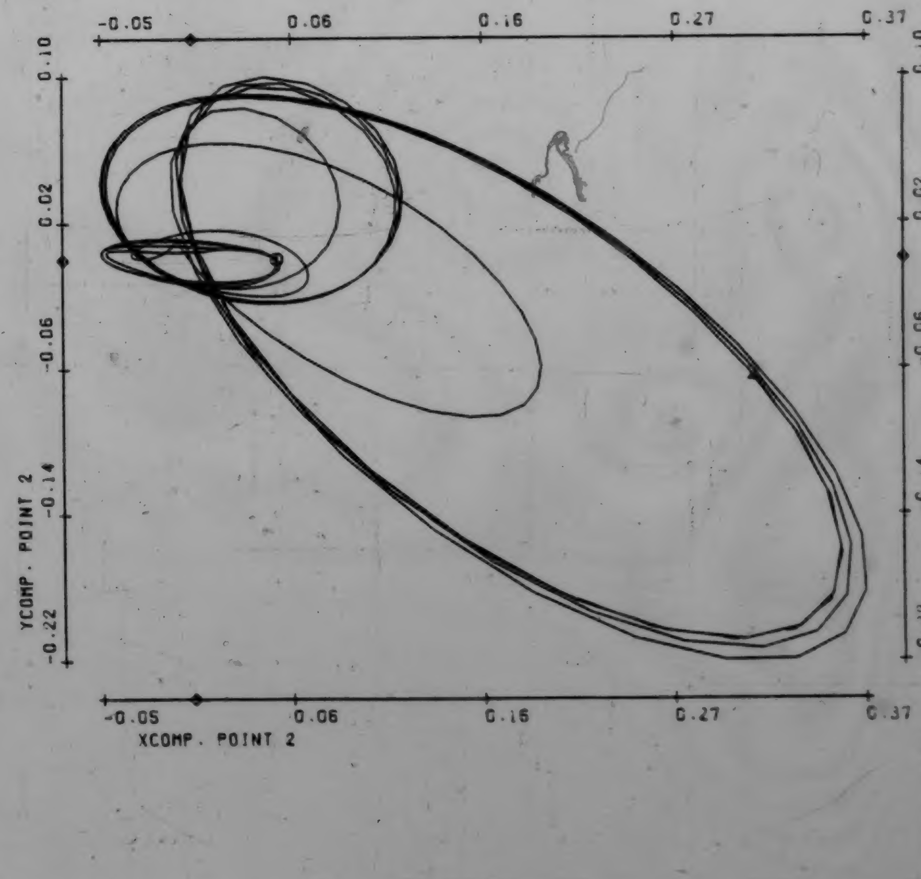

-1.19
-0.86
-0.53
-0.20
0.13
1.30
0.95
0.61
0.27
-0.07
ZCOMP. POINT 4
YCOMP. POINT 4

FIGURE T5

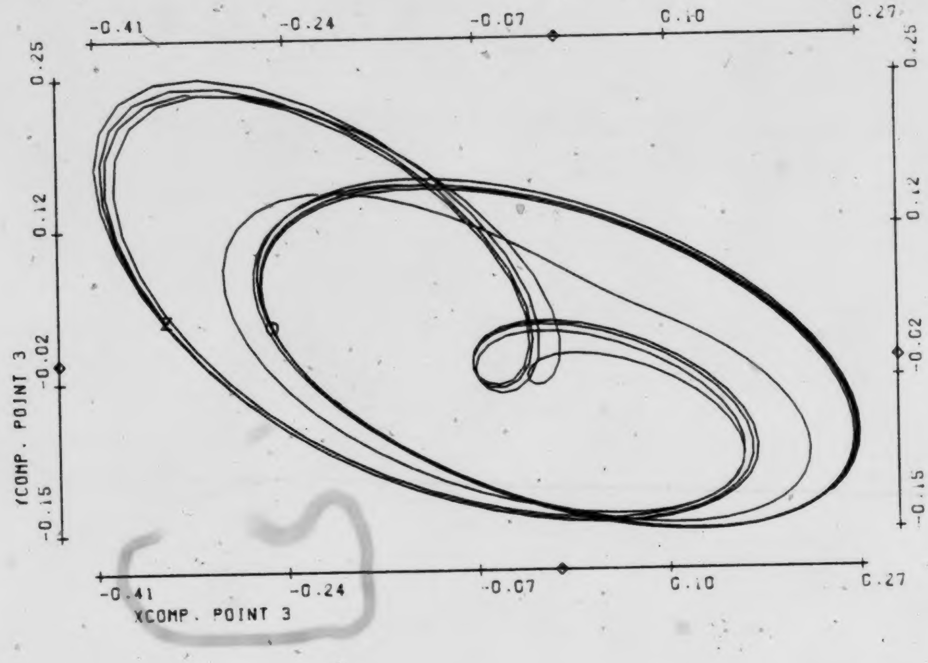
Bifurcation parameter η	.325
No. of timesteps	500
Stepsize	.03
Init. condition for $\vec{V}_2$	.001 (all)
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.66
Mean velocity:	
Point 1	1.95
2	1.64
3	2.65
4	1.60

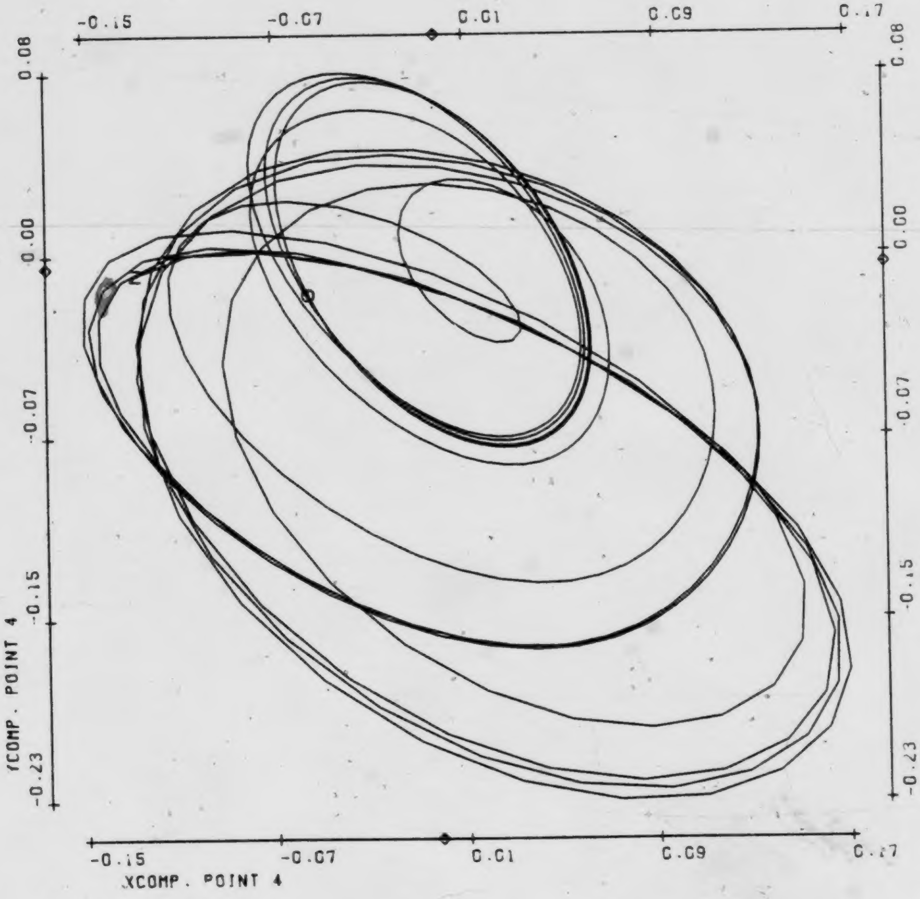
T5(1)


-0.46
-0.22
0.01
0.25
0.49
0.33
0.10
-0.13
-0.35
YCOMP. POINT 1
XCOMP. POINT 1

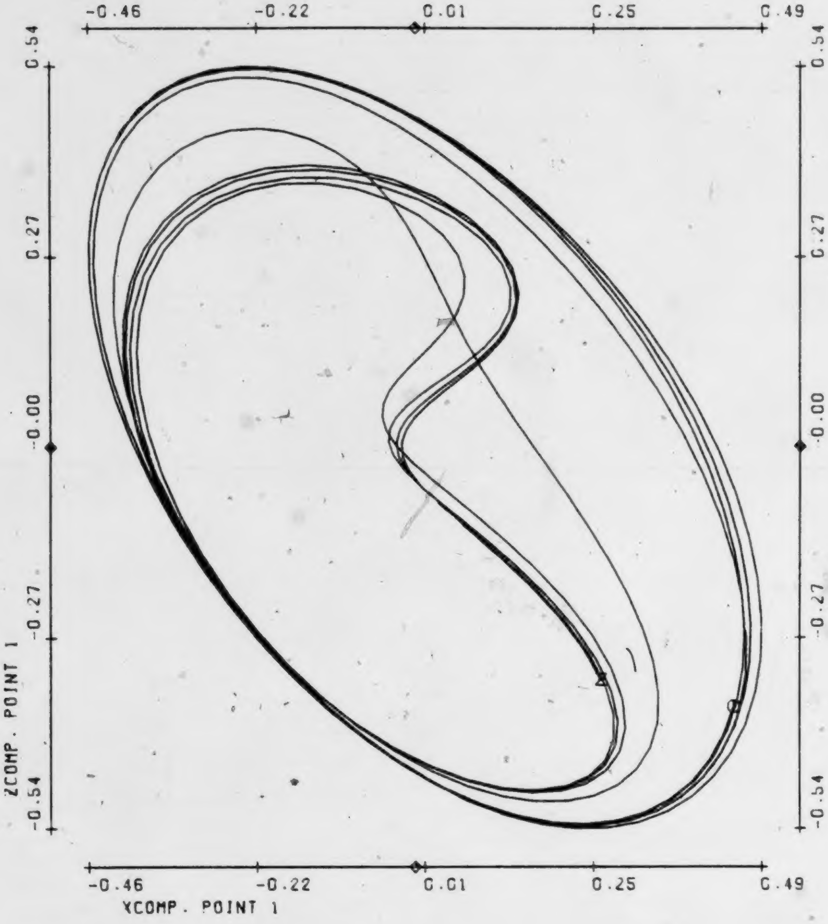

-0.05
0.06
0.16
0.27
0.37
0.10
0.02
-0.06
-0.14
-0.22
YCOMP. POINT 2
XCOMP. POINT 2

T5(2)

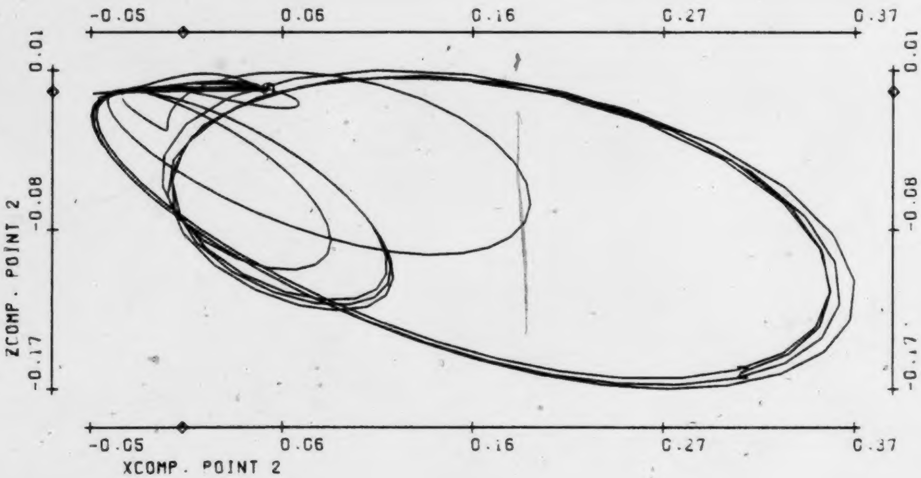

-0.41
-0.24
-0.07
0.10
0.27
0.25
0.12
-0.02
-0.15
YCOMP. POINT 3
XCOMP. POINT 3


-0.15
-0.07
0.01
0.09
0.17
0.08
0.00
-0.07
-0.15
-0.23
YCOMP. POINT 4
XCOMP. POINT 4

T5(3)

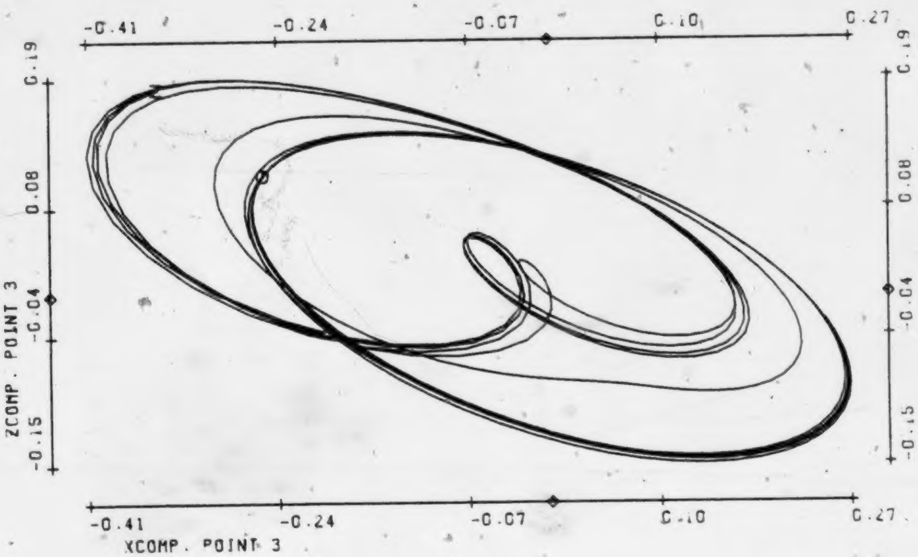


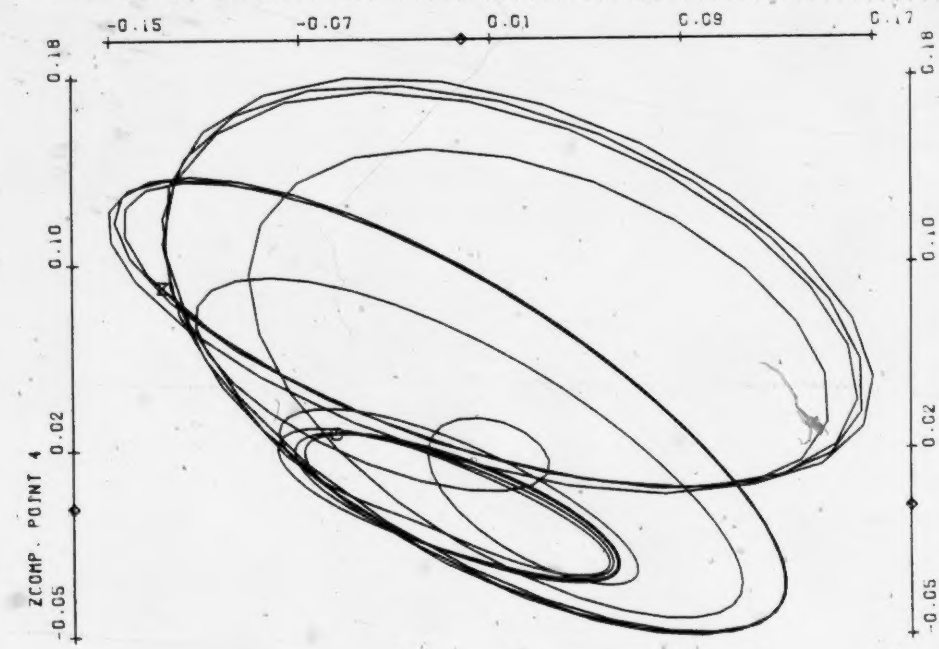
-0.46
-0.22
0.01
0.25
0.49
0.54
0.27
-0.00
-0.27
-0.54
ZCOMP. POINT 1
XCOMP. POINT 1

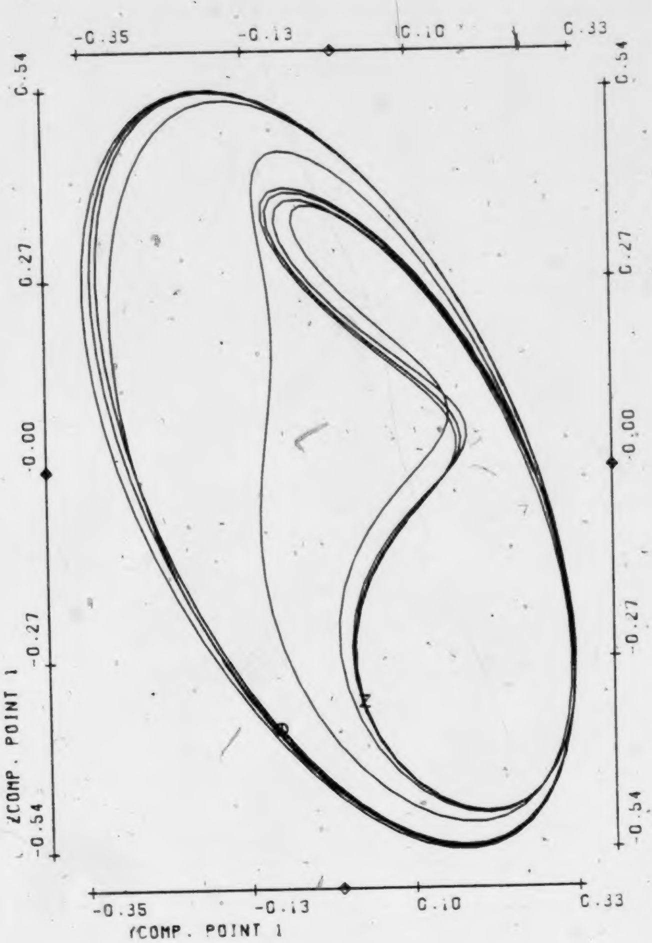


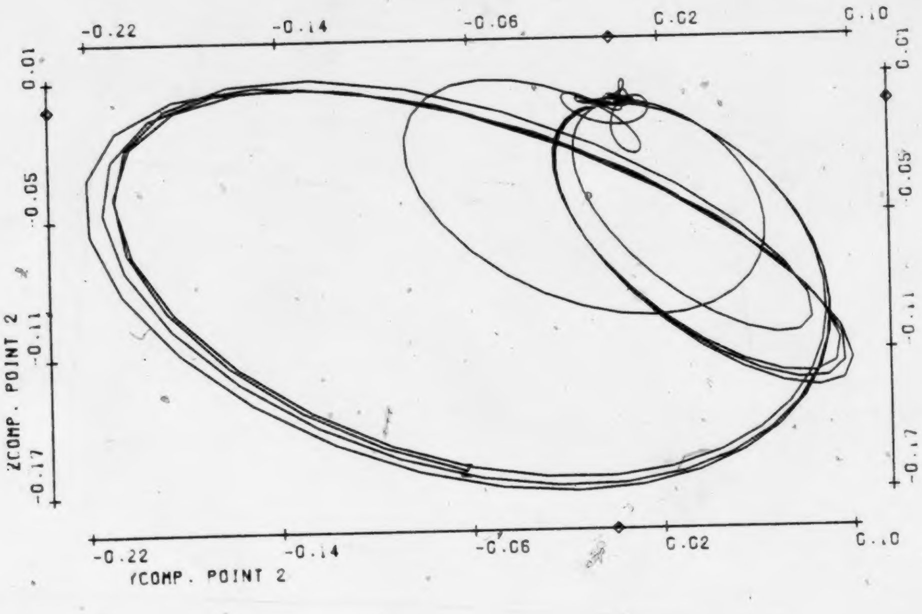
-0.05
0.06
0.16
0.27
0.37
0.01
-0.08
-0.17
ZCOMP. POINT 2
XCOMP. POINT 2

T5(4)

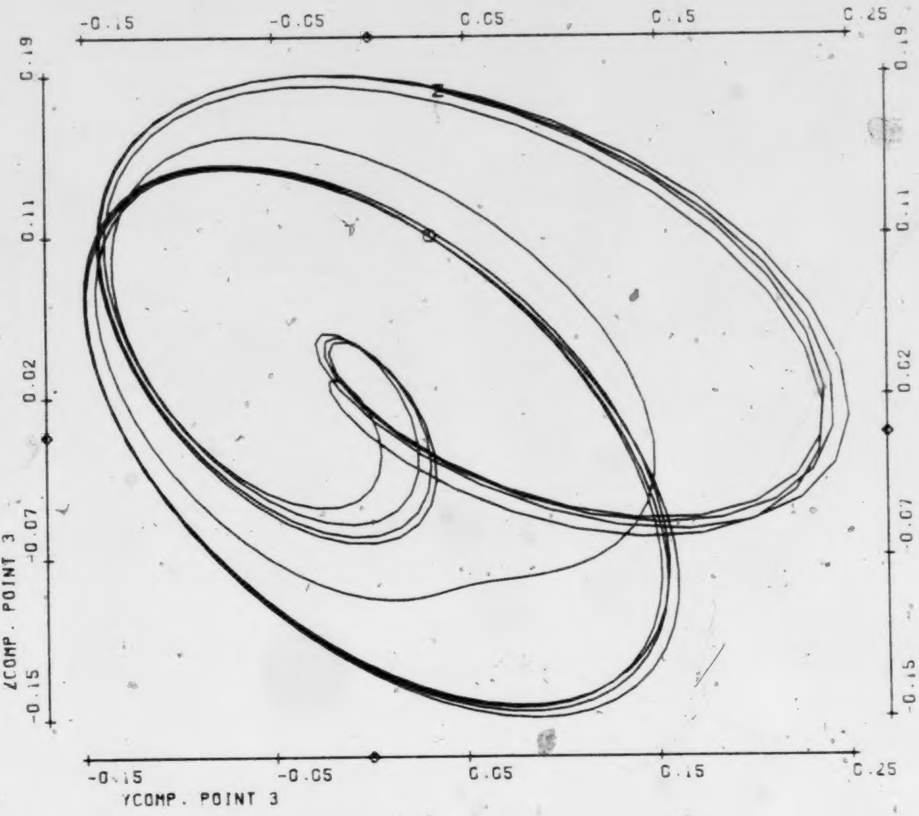

-0.41
-0.24
-0.07
0.10
0.27
0.19
0.08
-0.04
-0.15
ZCOMP. POINT 3
XCOMP. POINT 3


-0.15
-0.07
0.01
0.09
0.17
0.18
0.10
0.02
-0.05
ZCOMP. POINT 4
XCOMP. POINT 4


-0.35
-0.13
0.10
0.33
0.54
0.27
0.00
-0.27
-0.54
ZCOMP. POINT 1
YCOMP. POINT 1


-0.22
-0.14
-0.06
0.02
0.10
0.01
-0.05
-0.11
-0.17
ZCOMP. POINT 2
YCOMP. POINT 2

T5(6)


-0.15
-0.05
0.05
0.15
0.25
0.19
0.11
0.02
-0.07
-0.15
ZCOMP. POINT 3
YCOMP. POINT 3

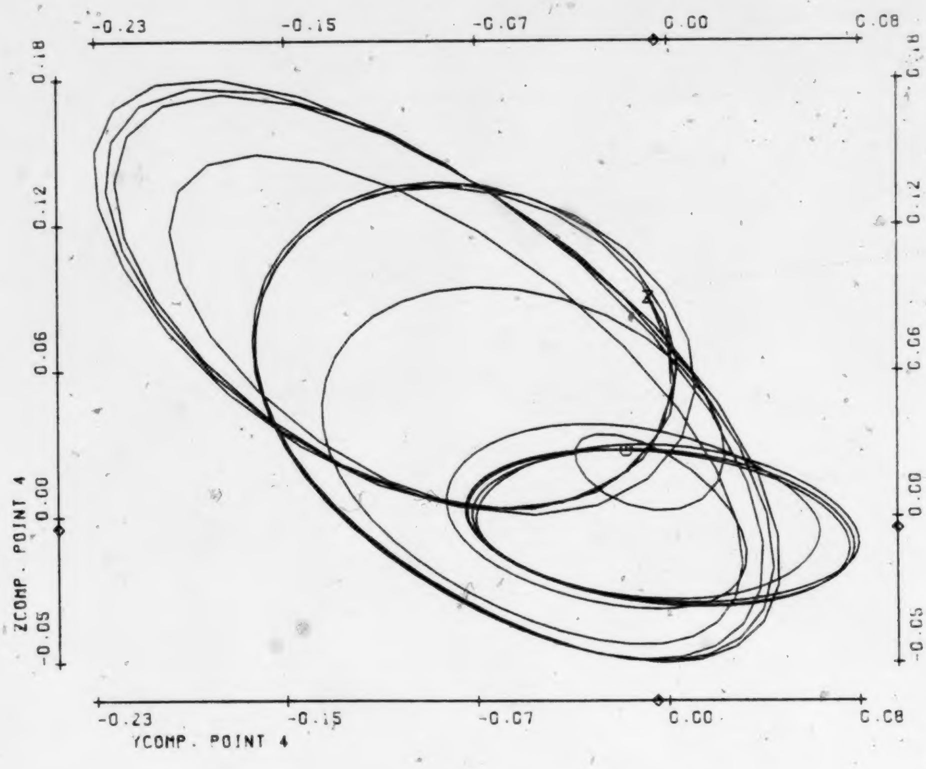
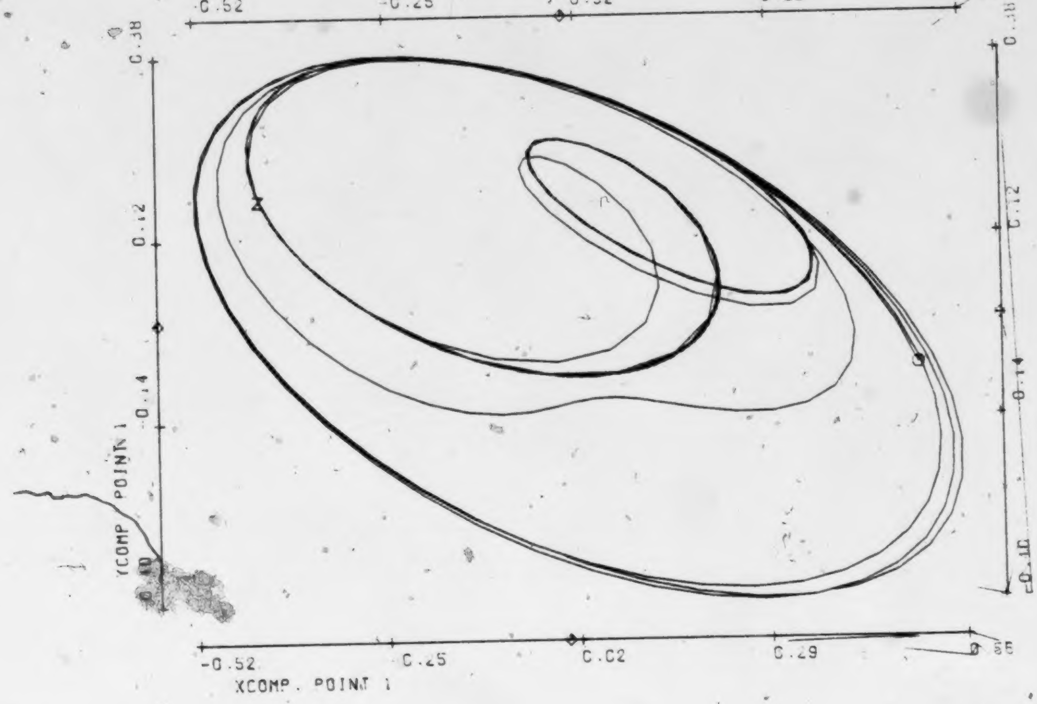
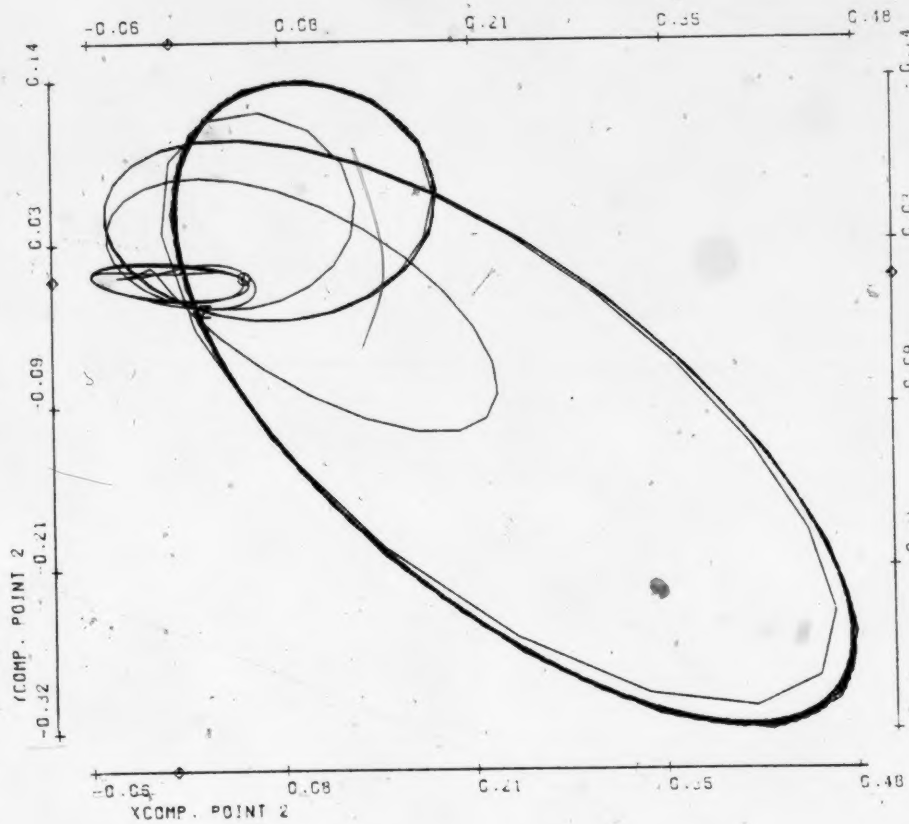

-0.23
-0.15
-0.07
0.00
0.08
0.18
0.12
0.06
0.00
-0.05
ZCOMP. POINT 4
YCOMP. POINT 4

FIGURE T6

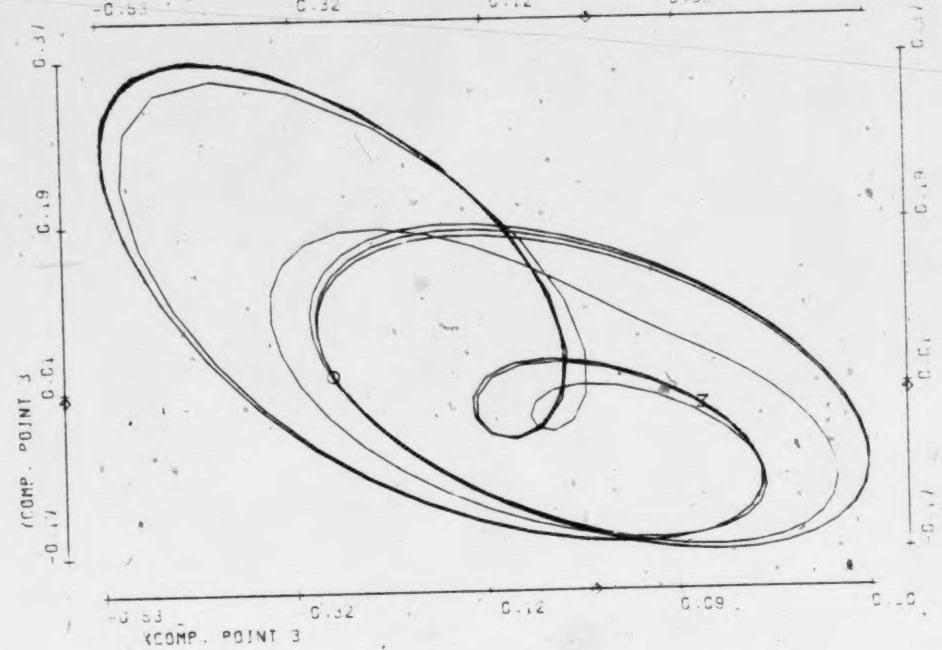
Bifurcation parameter η	.35
No. of timesteps	1000
Stepsize	.04
Init. condition for $\vec{V}_2$	.001 (X-comp.Pt.1) rest = 0
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.63
Mean velocity:	
Point 1	
2	
3	
4	

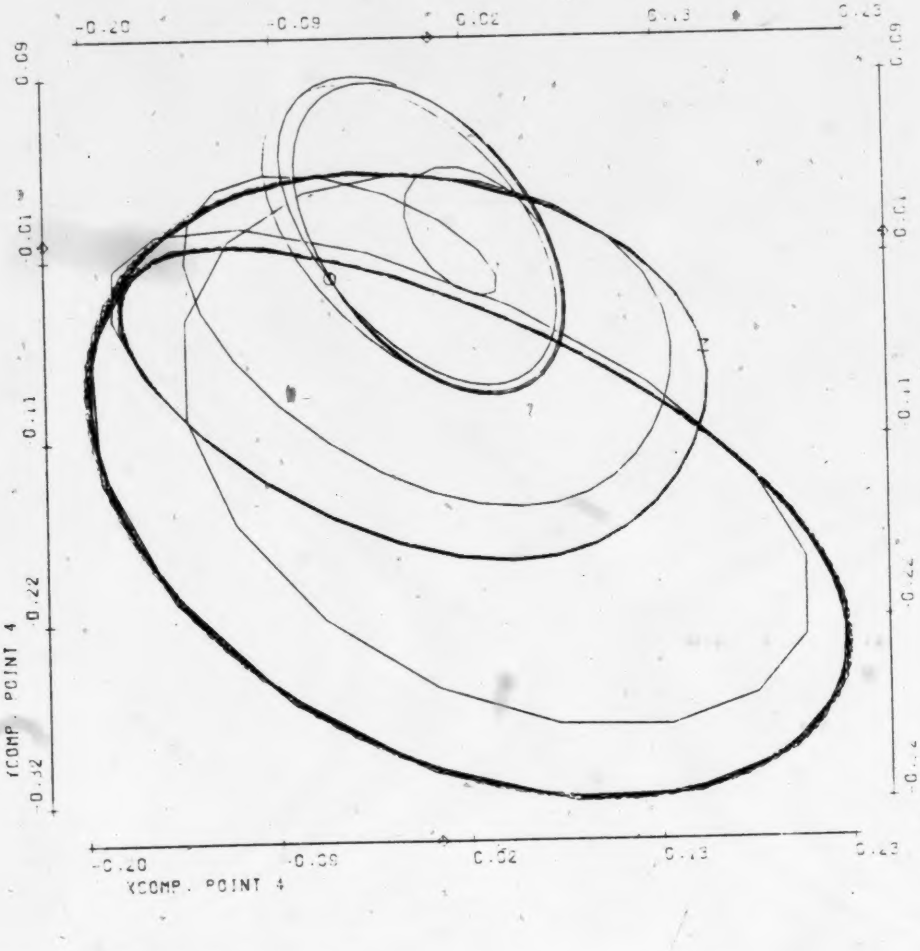
T6(1)

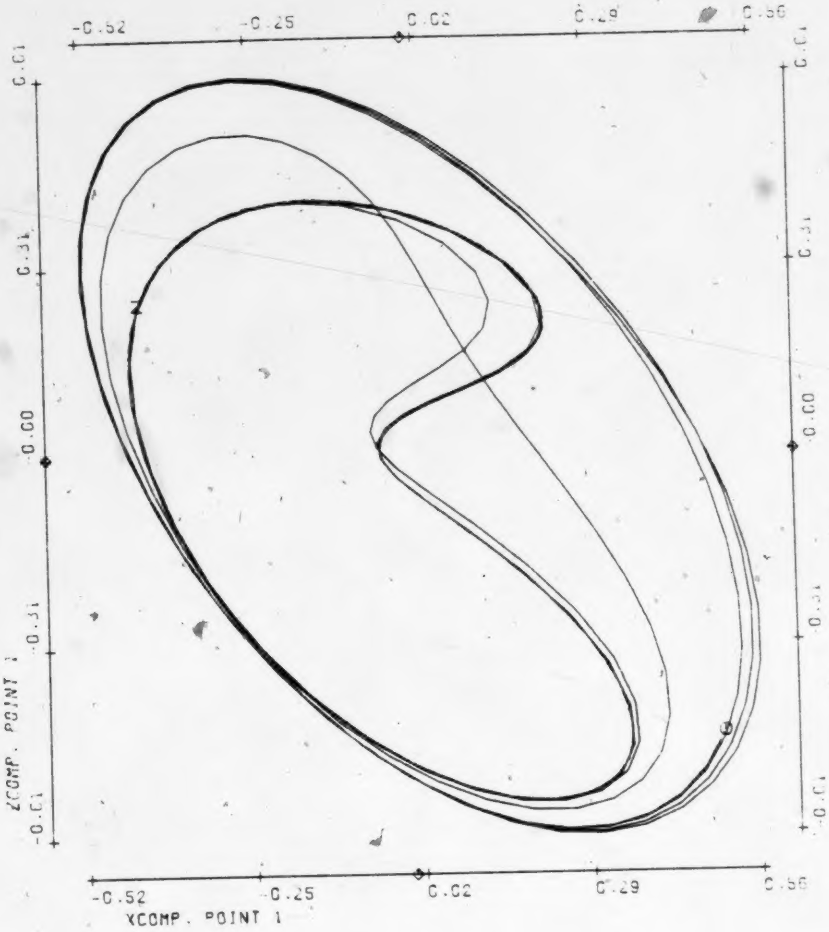

-0.52
-0.25
0.02
0.29
0.56
0.38
0.12
-0.14
-0.40
YCOMP. POINT 1
XCOMP. POINT 1

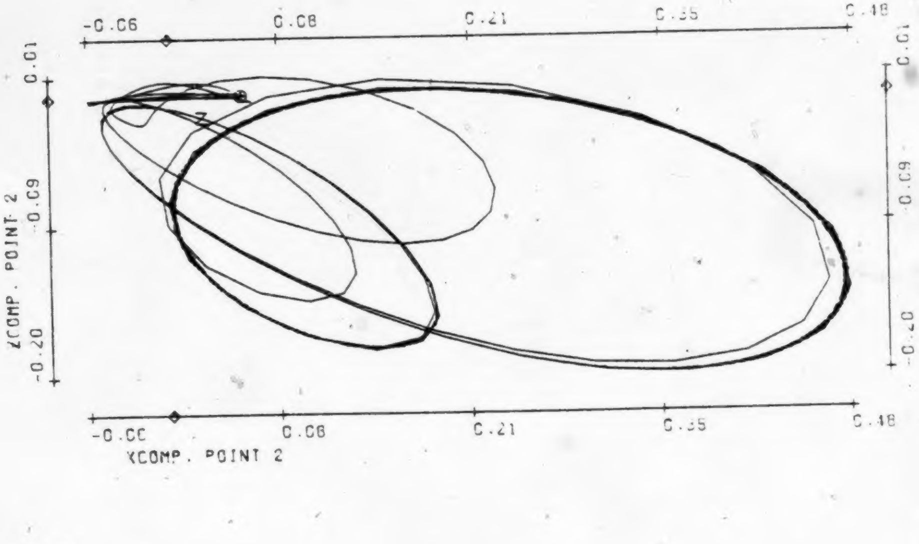

-0.06
0.08
0.21
0.35
0.48
0.14
0.03
-0.09
-0.21
-0.32
YCOMP. POINT 2
XCOMP. POINT 2

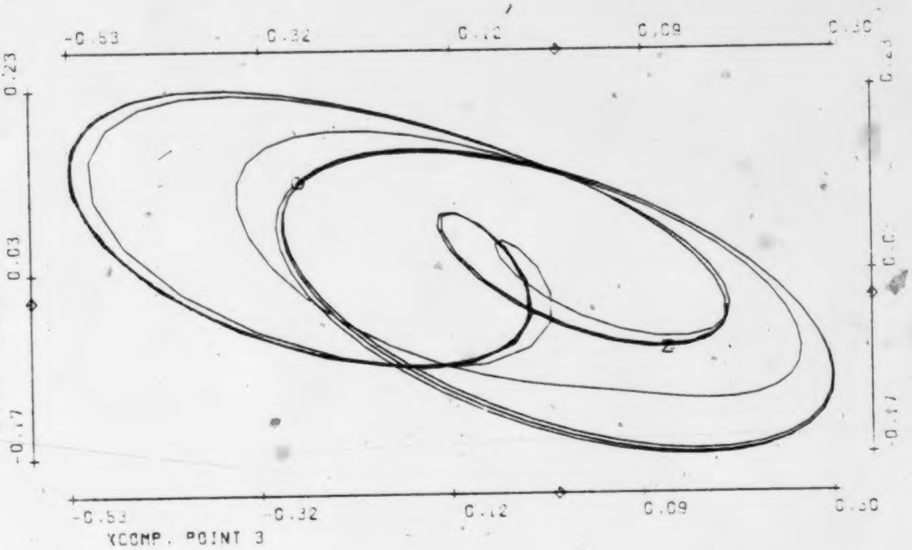
T6(2)

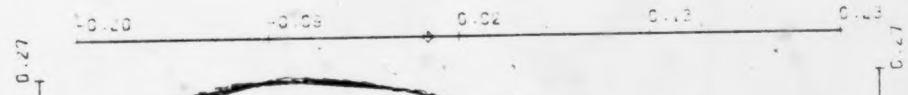

-0.53
-0.32
-0.12
0.09
0.30
0.37
0.19
0.01
-0.17
YCOMP. POINT 3
XCOMP. POINT 3

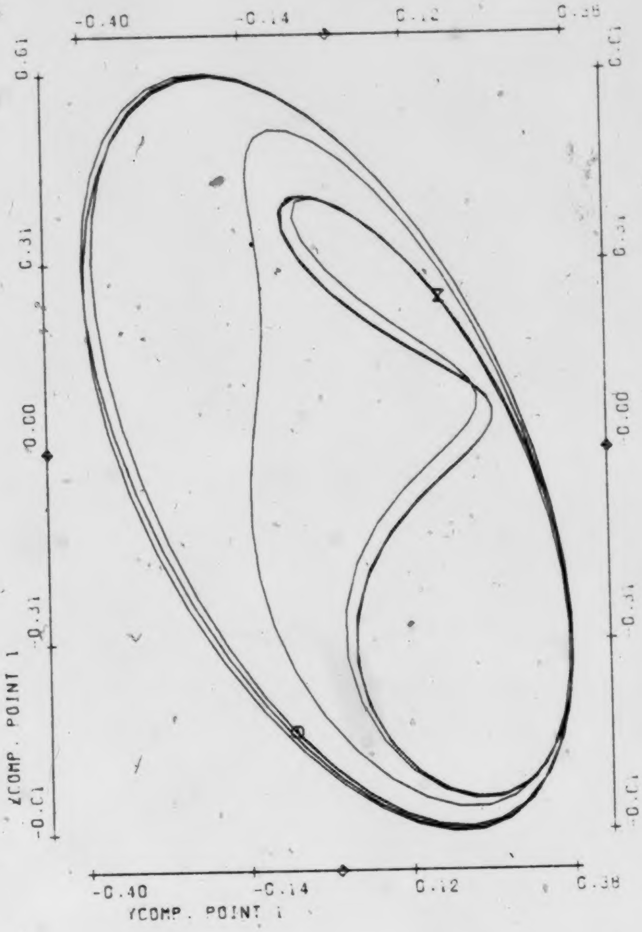

-0.20
-0.09
0.02
0.13
0.23
0.09
-0.01
-0.11
-0.22
-0.32
YCOMP. POINT 4
XCOMP. POINT 4

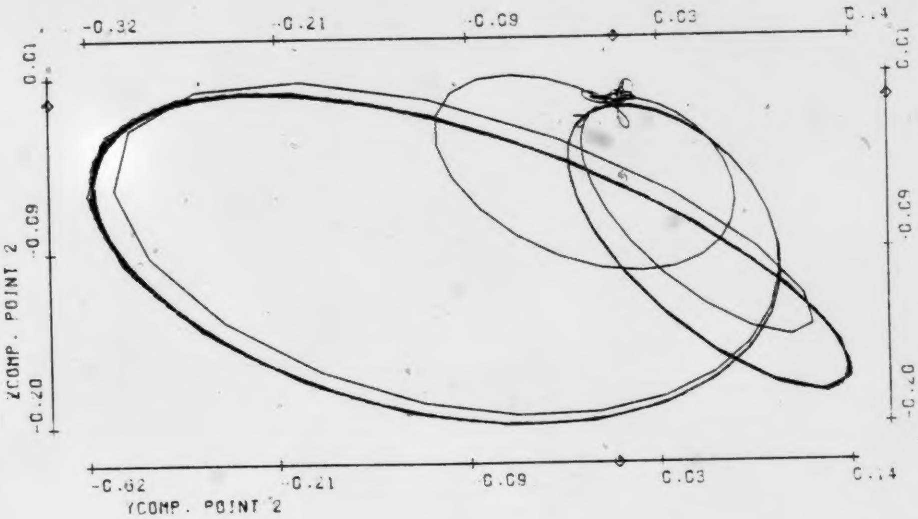

-0.52
-0.25
0.02
0.29
0.56
0.61
0.31
0.00
-0.31
-0.61
ZCOMP. POINT 1
XCOMP. POINT 1

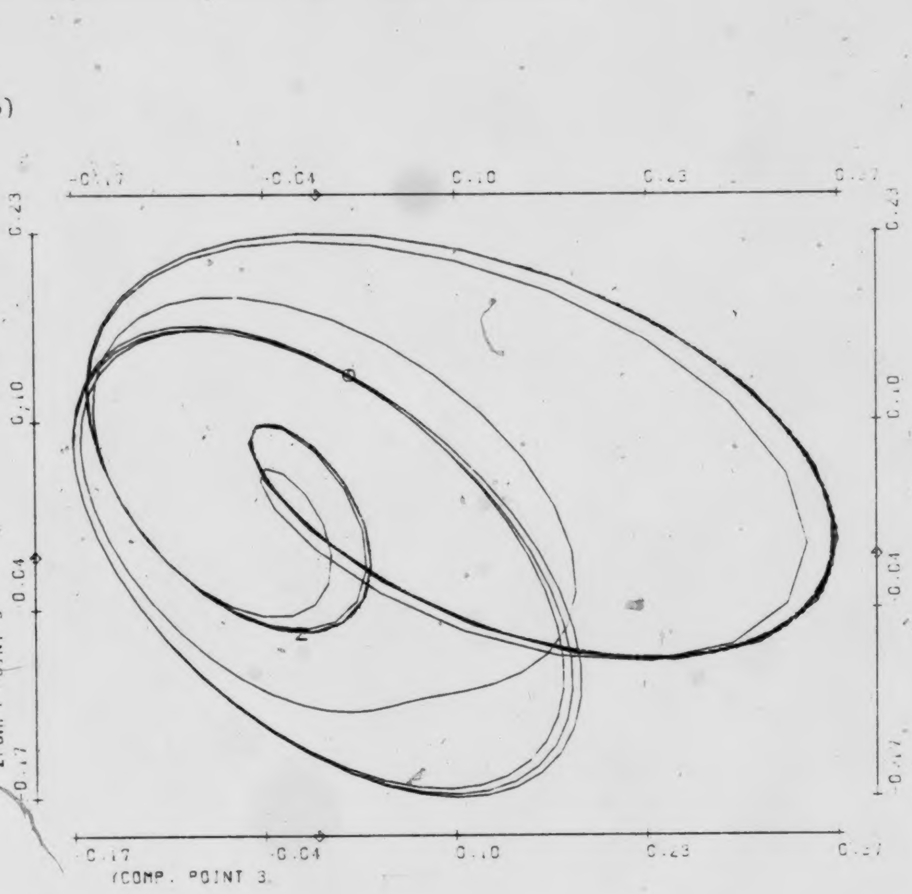

-0.06
0.08
0.21
0.35
0.48
0.01
-0.09
-0.20
ZCOMP. POINT 2
XCOMP. POINT 2


-0.53
-0.32
-0.12
0.09
0.30
0.23
0.03
-0.17
ZCOMP. POINT 3
XCOMP. POINT 3


-0.20
-0.09
0.02
0.13
0.23
0.27
0.19
0.11
0.03
-0.06
ZCOMP. POINT 4
XCOMP. POINT 4


-0.40
-0.14
0.12
0.38
0.61
0.31
0.00
-0.31
-0.61
ZCOMP. POINT 1
YCOMP. POINT 1


-0.32
-0.21
-0.09
0.03
0.14
0.01
-0.09
-0.20
ZCOMP. POINT 2
YCOMP. POINT 2


-0.17
-0.04
0.10
0.23
0.37
0.23
0.10
-0.04
-0.17
ZCOMP. POINT 3
YCOMP. POINT 3

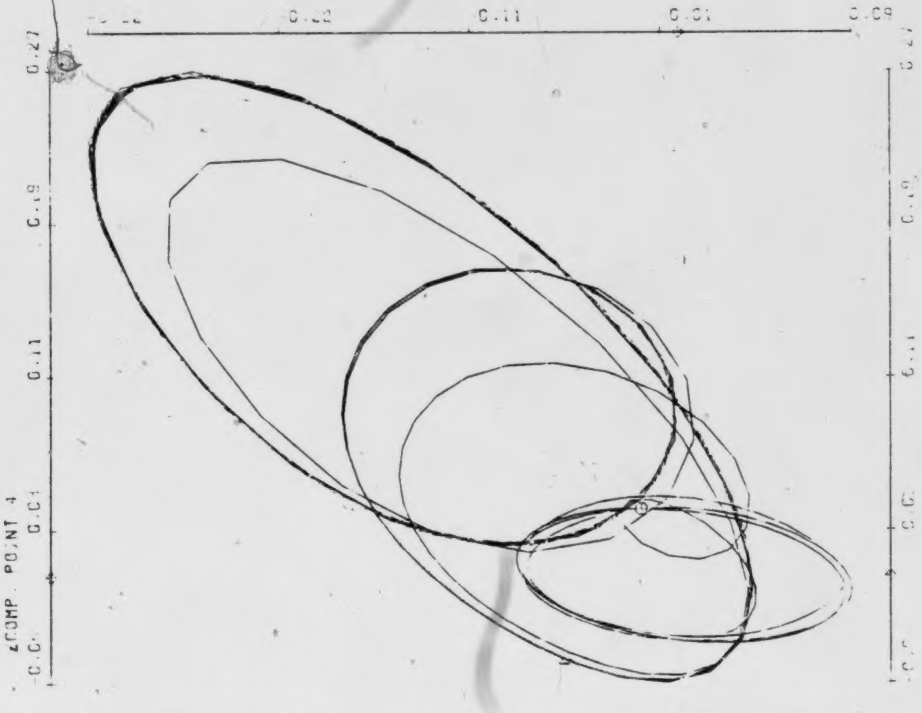
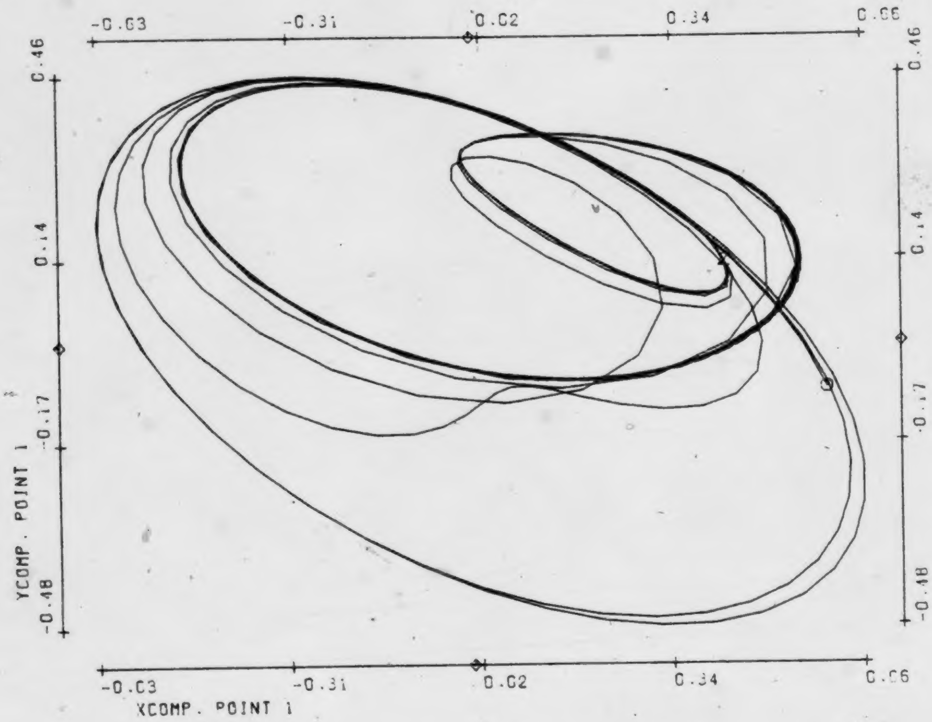
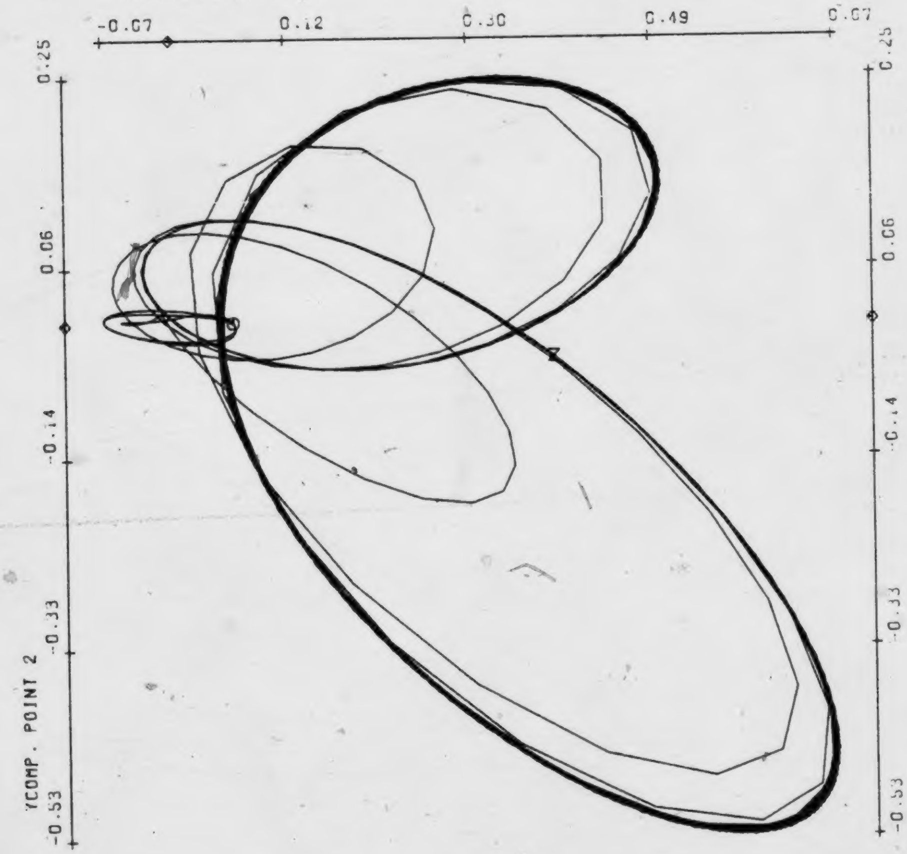

-0.32
-0.22
-0.11
-0.01
0.09
0.27
0.19
0.11
0.03
-0.0
ZCOMP. POINT 4
YCOMP. POINT 1

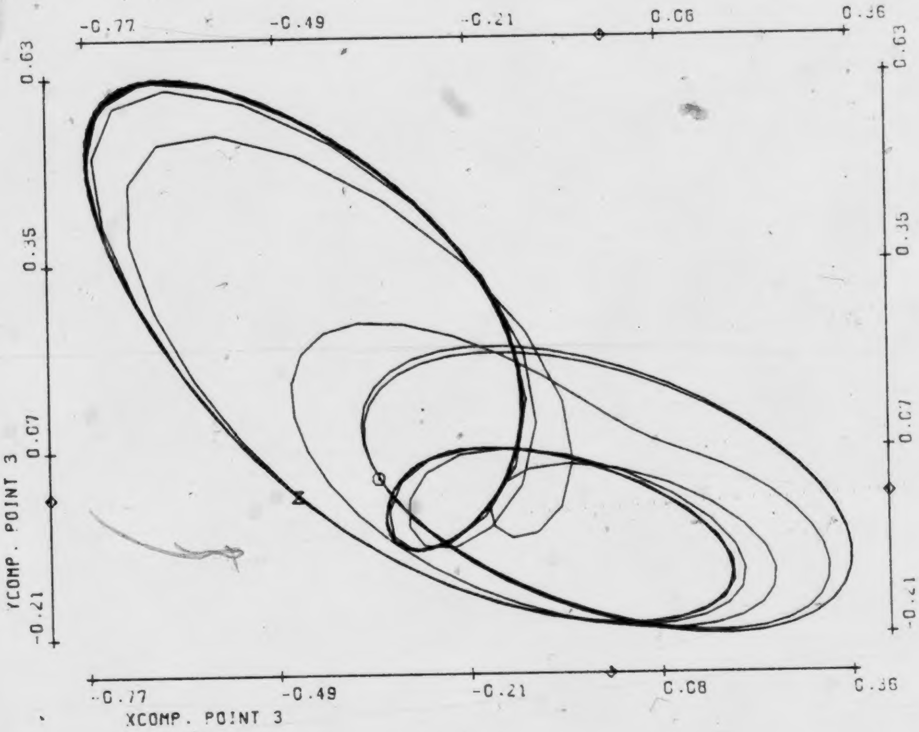
FIGURE T7

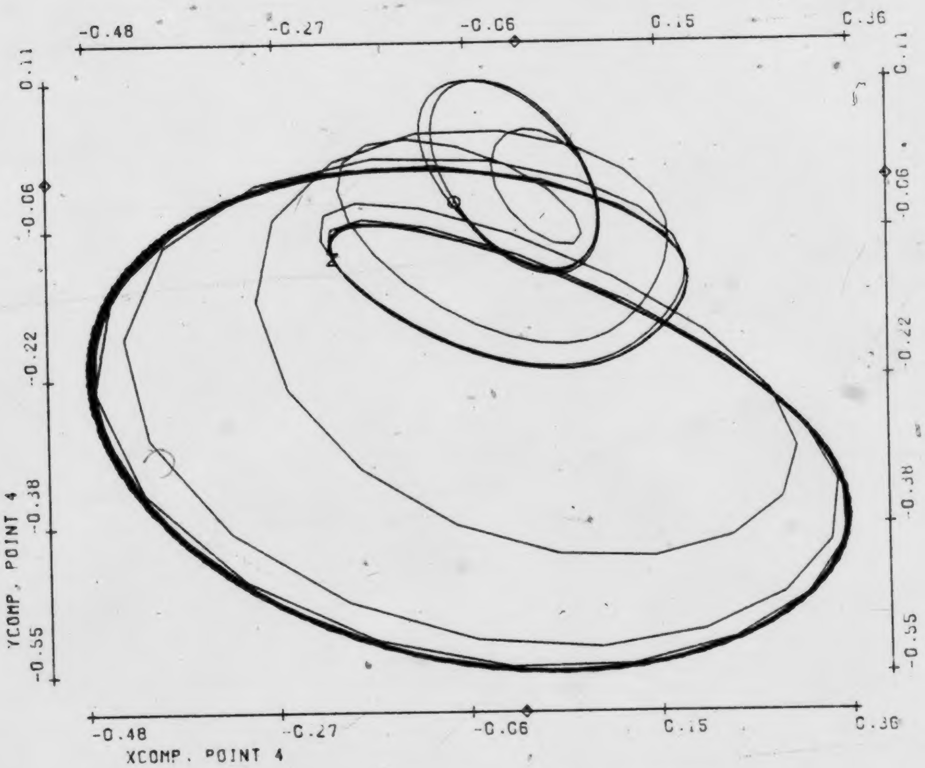
Bifurcation parameter η	.4
No. of timesteps	800
Stepsize	.04
Init. condition for $\vec{V}_2$	.001 (X-comp.Pt.1) rest = 0
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.60
Mean velocity:	
Point 1	1.56
2	1.61
3	2.61
4	1.73

T7(1)

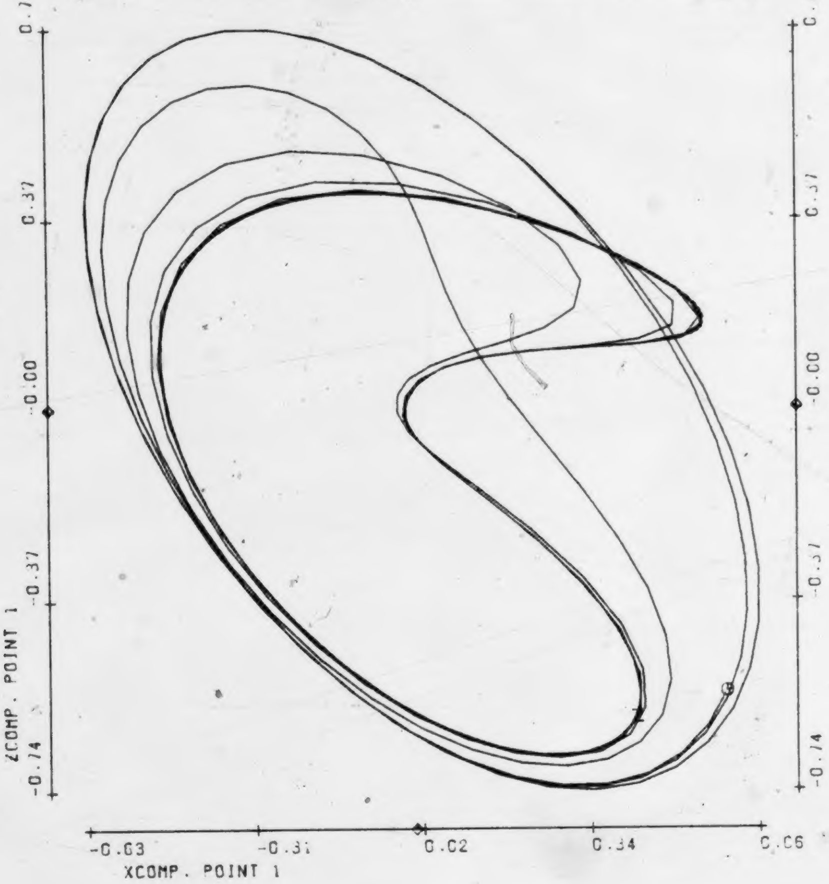

-0.63
-0.31
0.02
0.34
0.66
0.46
0.14
-0.17
-0.48
YCOMP. POINT 1
XCOMP. POINT 1

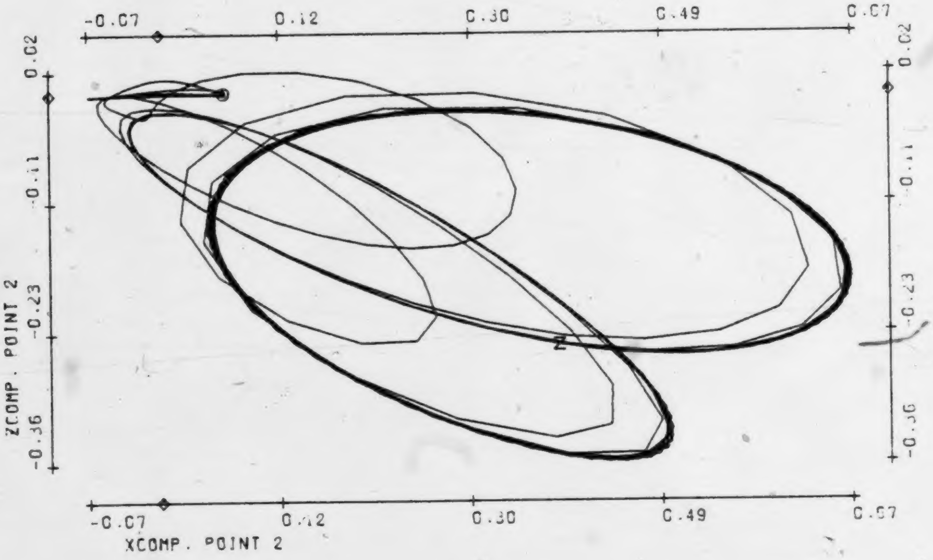

-0.07
0.12
0.30
0.49
0.67
0.25
0.06
-0.14
-0.33
-0.53
YCOMP. POINT 2
XCOMP. POINT 2


-0.77
-0.49
-0.21
0.08
0.36
0.63
0.35
0.07
-0.21
YCOMP. POINT 3
XCOMP. POINT 3

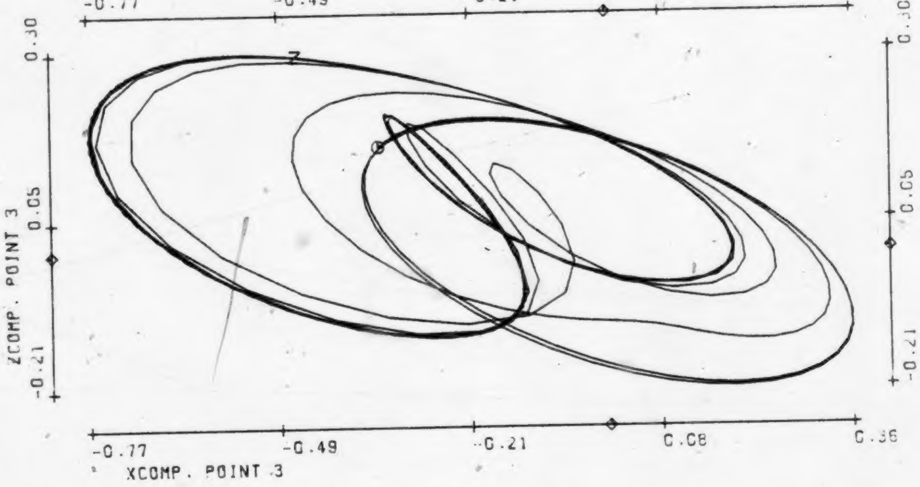

-0.48
-0.27
-0.06
0.15
0.36
0.11
-0.06
-0.22
-0.38
-0.55
YCOMP. POINT 4
XCOMP. POINT 4

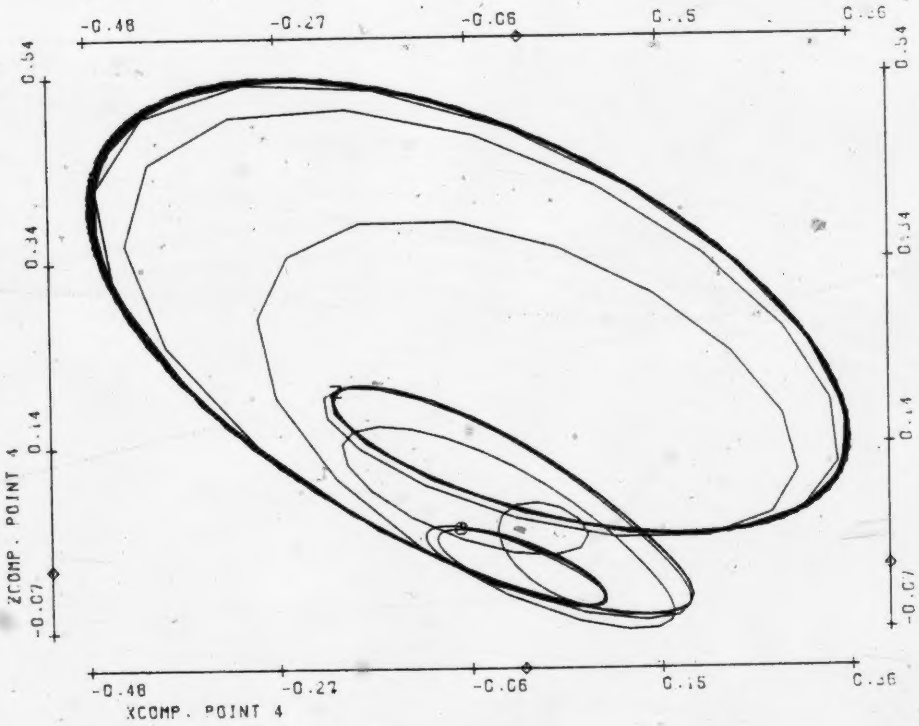
T7(3)

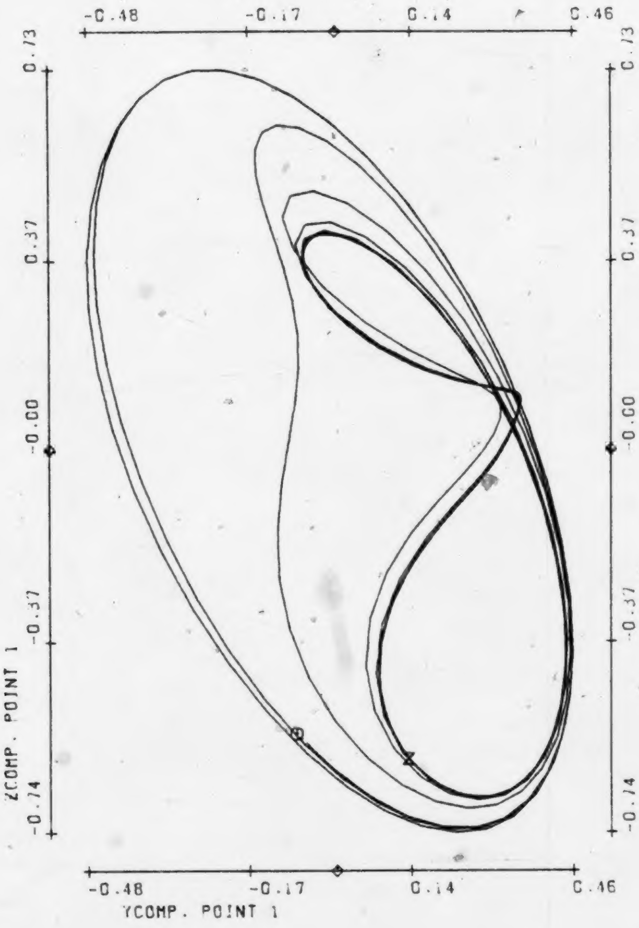

-0.63
-0.31
0.02
0.34
0.66
0.73
0.37
-0.00
-0.37
-0.74
ZCOMP. POINT 1
XCOMP. POINT 1

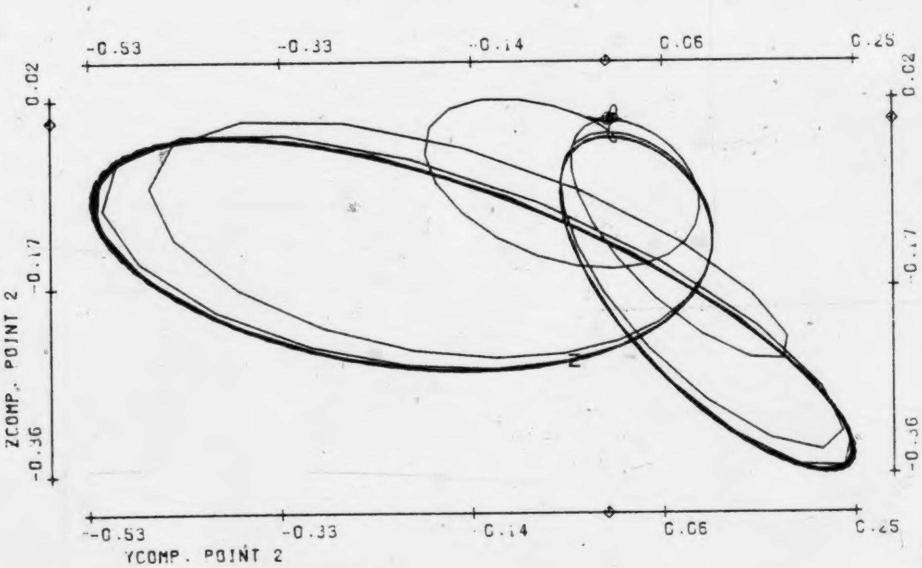

-0.07
0.12
0.30
0.49
0.67
0.02
-0.11
-0.23
-0.36
Z
ZCOMP. POINT 2
XCOMP. POINT 2

T7(4)

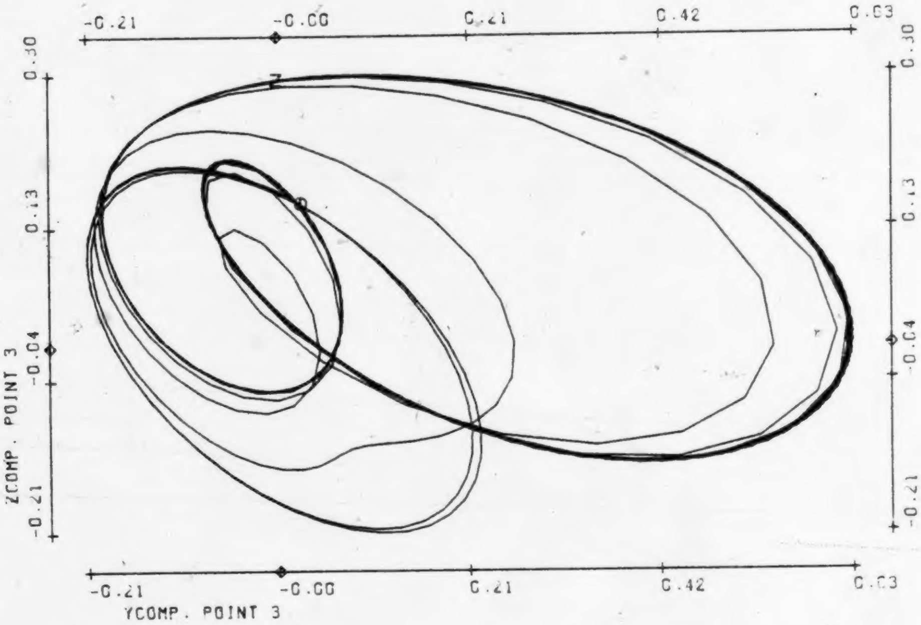

-0.77
-0.49
-0.21
0.08
0.36
0.30
0.05
-0.21
ZCOMP. POINT 3
XCOMP. POINT 3


-0.48
-0.27
-0.06
0.15
0.36
0.54
0.34
0.14
-0.07
ZCOMP. POINT 4
XCOMP. POINT 4


-0.48
-0.17
0.14
0.46
0.73
0.37
-0.00
-0.37
-0.74
ZCOMP. POINT 1
YCOMP. POINT 1


-0.53
-0.33
-0.14
0.06
0.25
0.02
-0.17
-0.36
ZCOMP. POINT 2
YCOMP. POINT 2

T7(6)


-0.21
-0.00
0.21
0.42
0.63
0.30
0.13
-0.04
-0.21
Z
ZCOMP. POINT 3
YCOMP. POINT 3

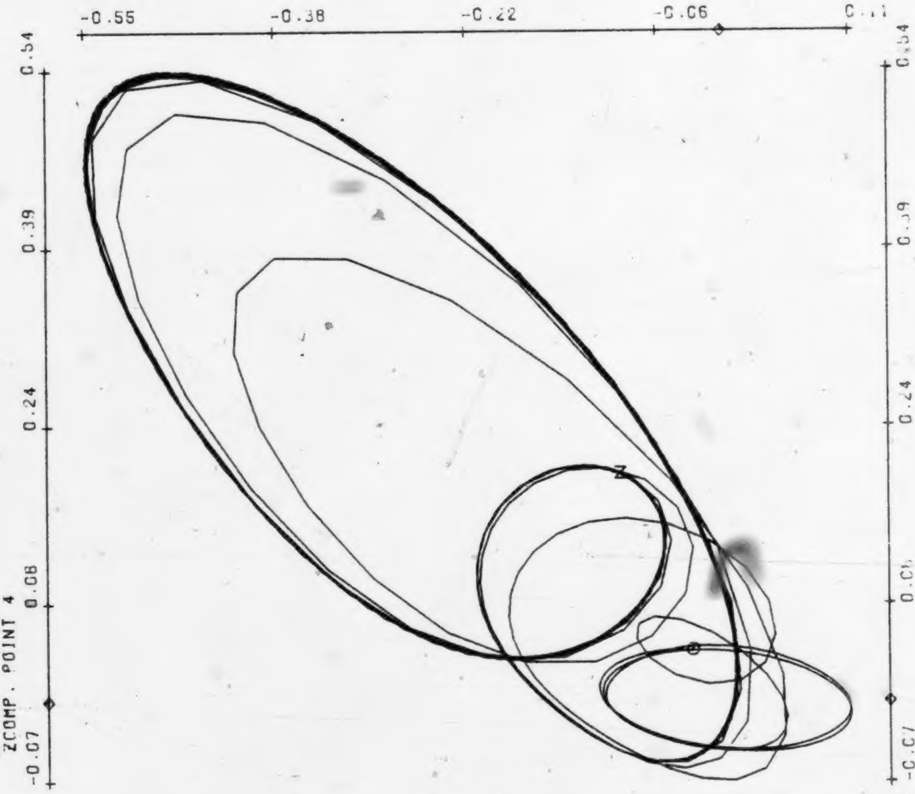
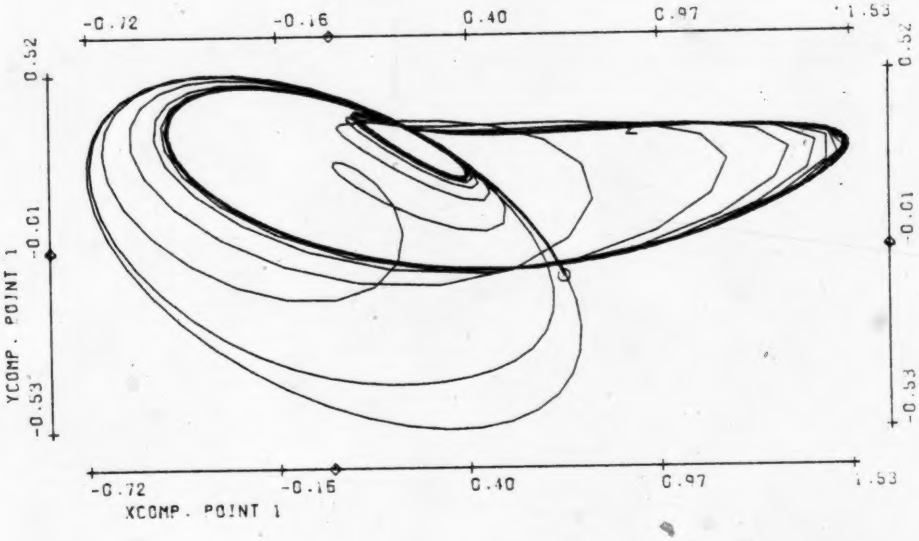
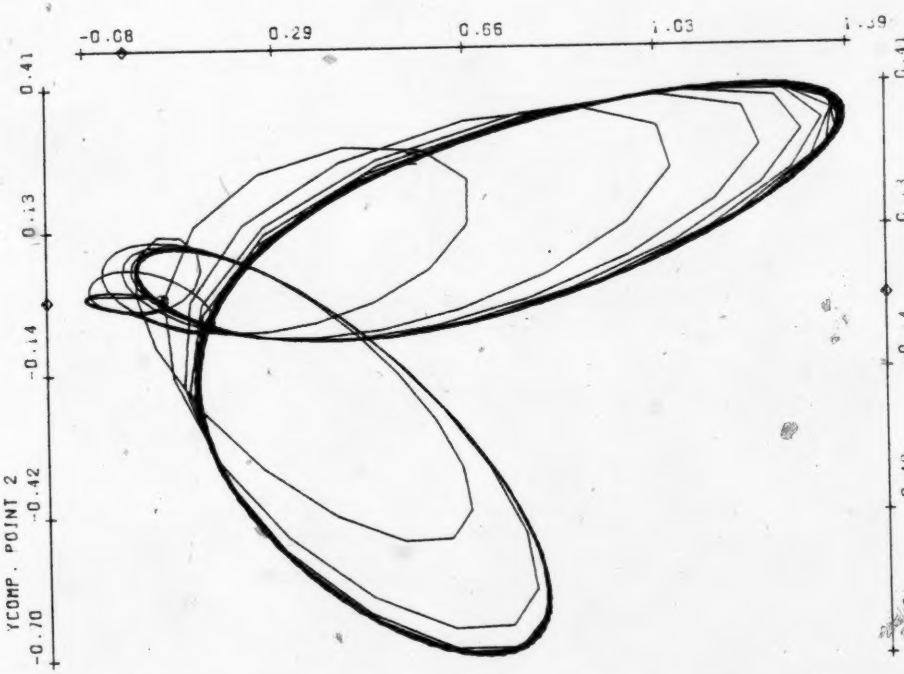

-0.55
-0.38
-0.22
-0.06
0.11
0.54
0.39
0.24
0.08
-0.07
Z
ZCOMP. POINT 4
YCOMP. POINT 4

FIGURE T8

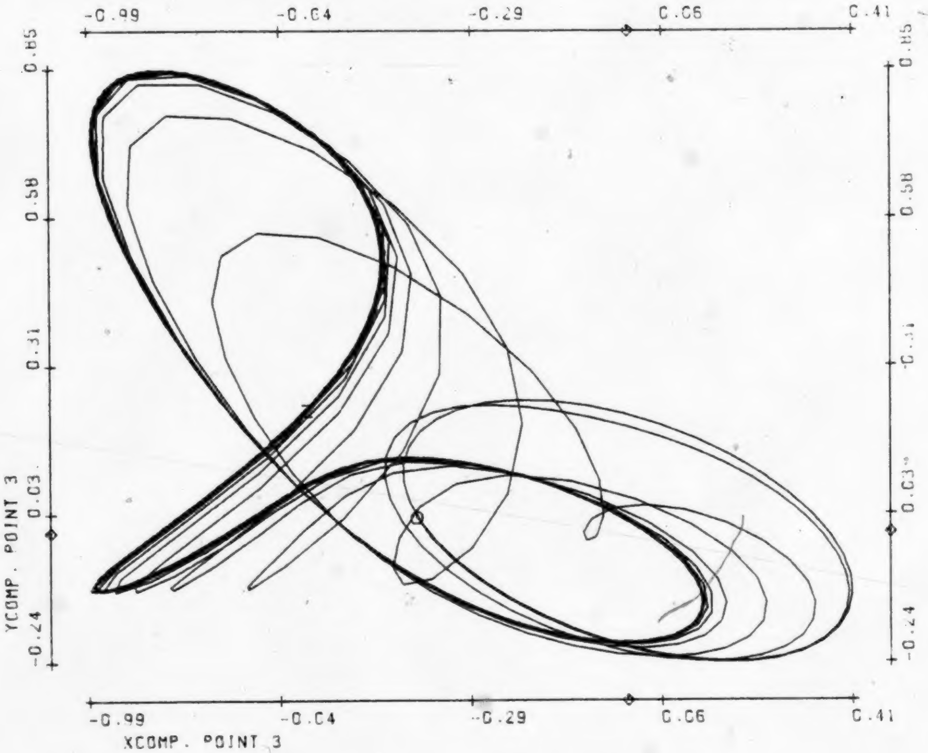
Bifurcation parameter η	.447
No. of timesteps	1000
Stepsize	.04
Init. condition for $\vec{V}_2$	.001(X-comp.Pt.1) rest = 0
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.55
Mean velocity:	
Point 1	1.39
2	1.73
3	2.81
4	1.91

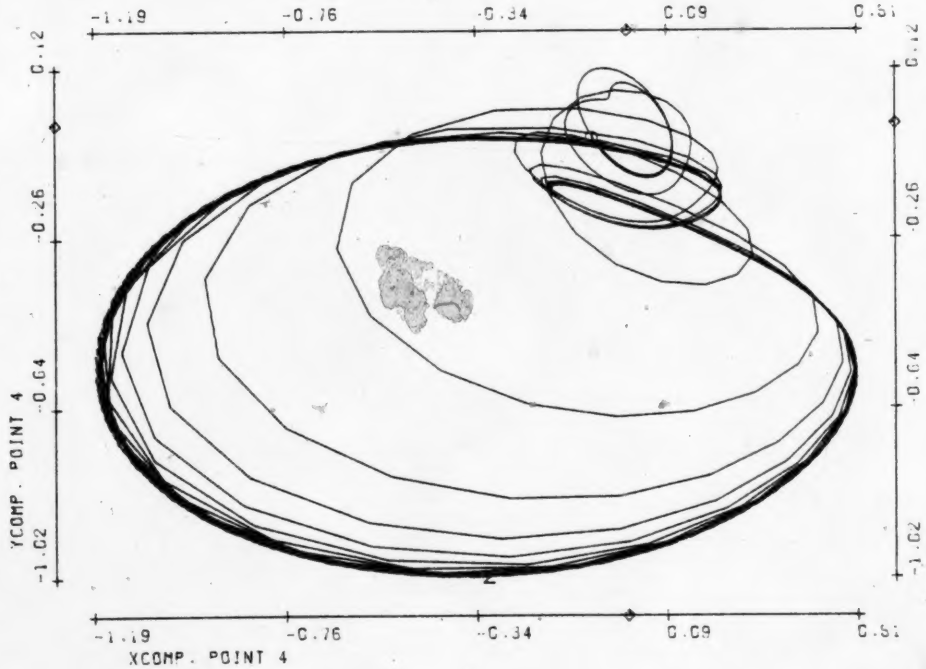
T8(1)


-0.72
-0.16
0.40
0.97
1.53
0.52
-0.01
-0.53
YCOMP. POINT 1
XCOMP. POINT 1

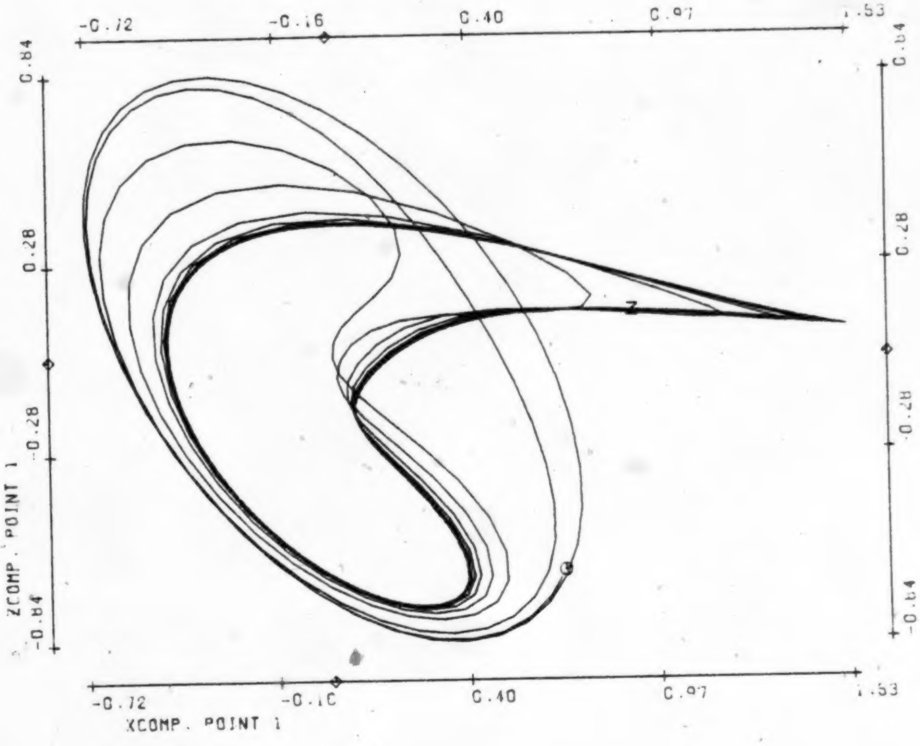

-0.08
0.29
0.66
1.03
1.39
0.41
0.13
-0.14
-0.42
-0.70
YCOMP. POINT 2
XCOMP. POINT 2

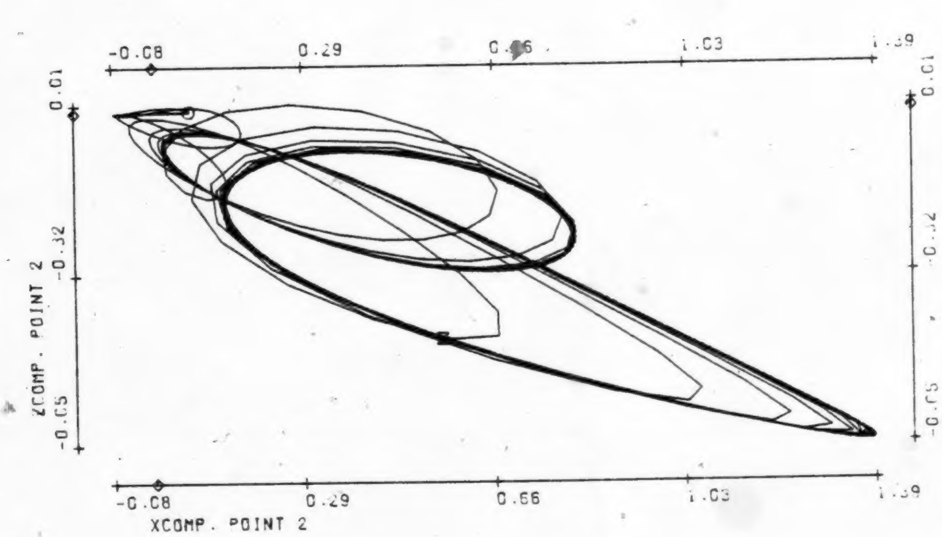
T8(2)

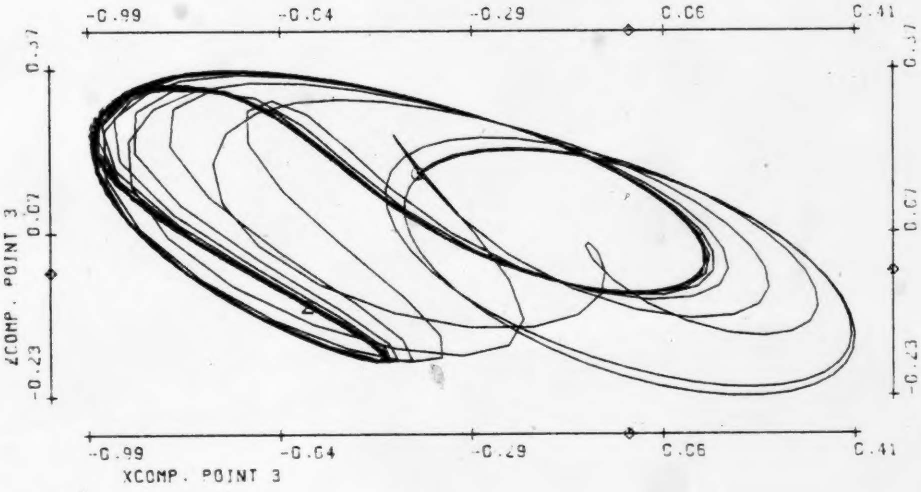

-0.99
-0.64
-0.29
0.06
0.41
0.85
0.58
0.31
0.03
-0.24
YCOMP. POINT 3
XCOMP. POINT 3

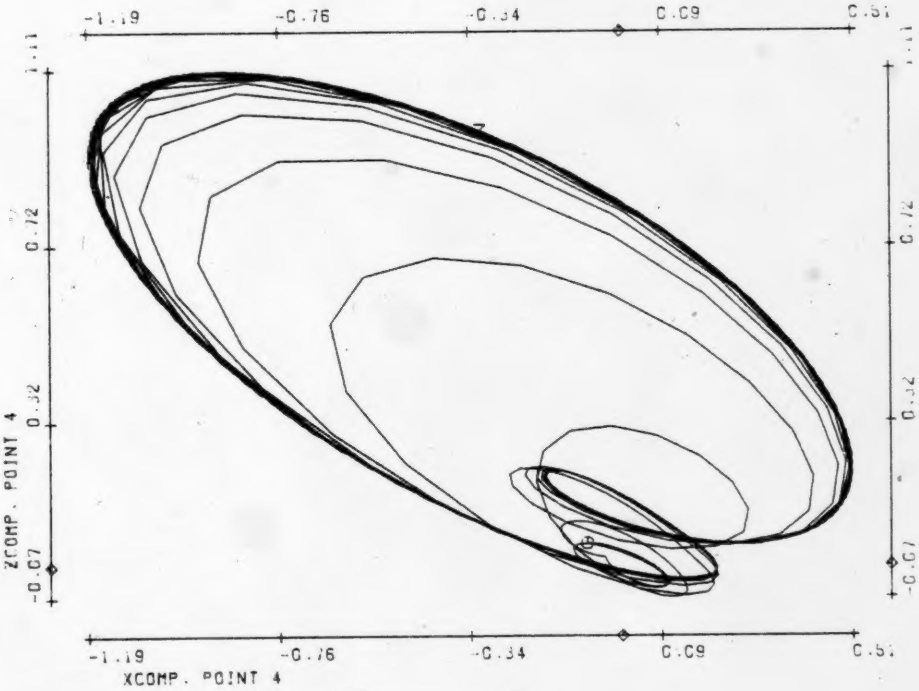

-1.19
-0.76
-0.34
0.09
0.51
0.12
-0.26
-0.64
-1.02
YCOMP. POINT 4
XCOMP. POINT 4

T8(3)

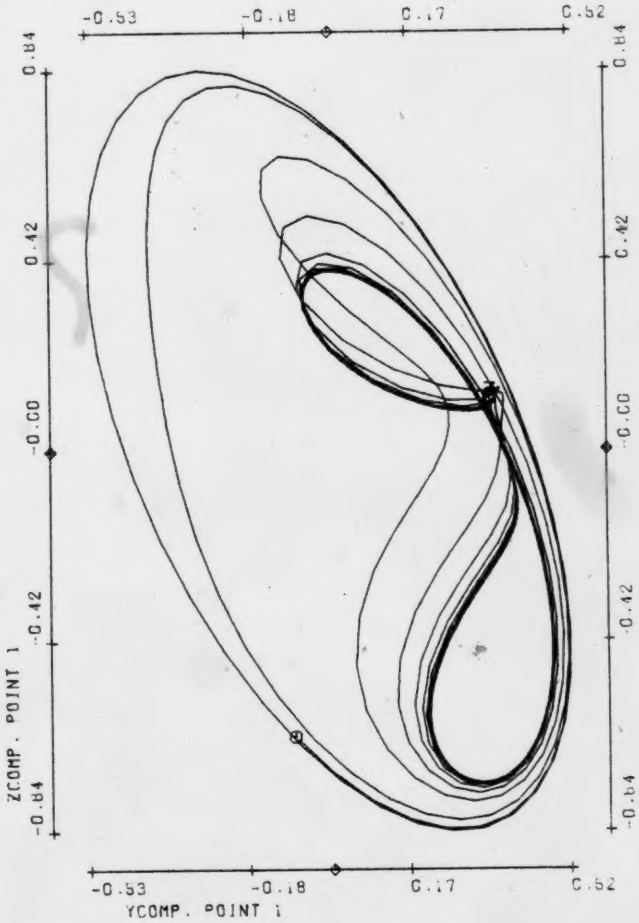

-0.72
-0.16
0.40
0.97
1.53
0.84
0.28
-0.28
-0.84
Z
ZCOMP. POINT 1
XCOMP. POINT 1

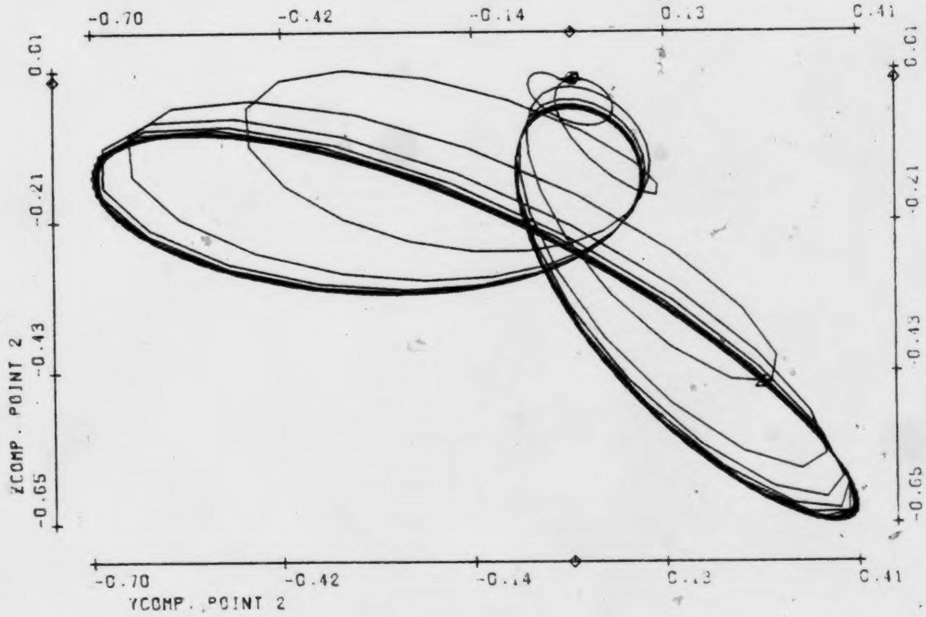

-0.08
0.29
0.66
1.03
1.39
0.01
-0.32
-0.65
Z
ZCOMP. POINT 2
XCOMP. POINT 2

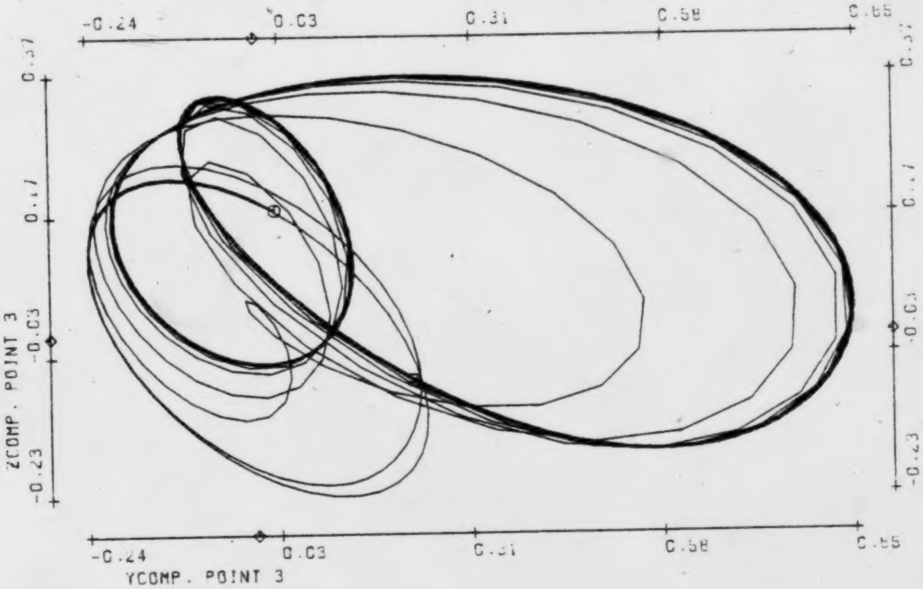

-0.99
-0.64
-0.29
0.06
0.41
0.37
0.07
-0.23
ZCOMP. POINT 3
XCOMP. POINT 3


-1.19
-0.76
-0.34
0.09
0.51
1.11
0.72
0.32
-0.07
ZCOMP. POINT 4
XCOMP. POINT 4

T8(5)


-0.53
-0.18
0.17
0.52
0.84
0.42
-0.00
-0.42
-0.84
ZCOMP. POINT 1
YCOMP. POINT 1


-0.70
-0.42
-0.14
0.13
0.41
0.01
-0.21
-0.43
-0.65
ZCOMP. POINT 2
YCOMP. POINT 2


-0.24
0.03
0.31
0.58
0.85
0.37
0.17
-0.03
-0.23
ZCOMP. POINT 3
YCOMP. POINT 3

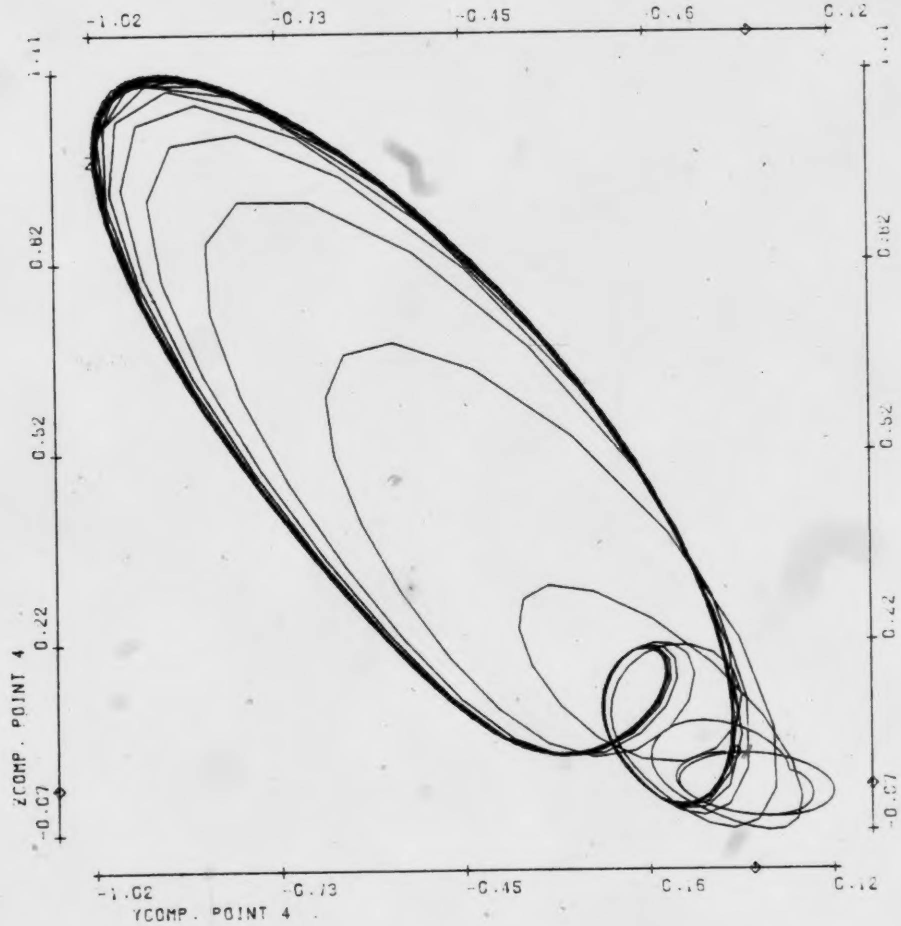
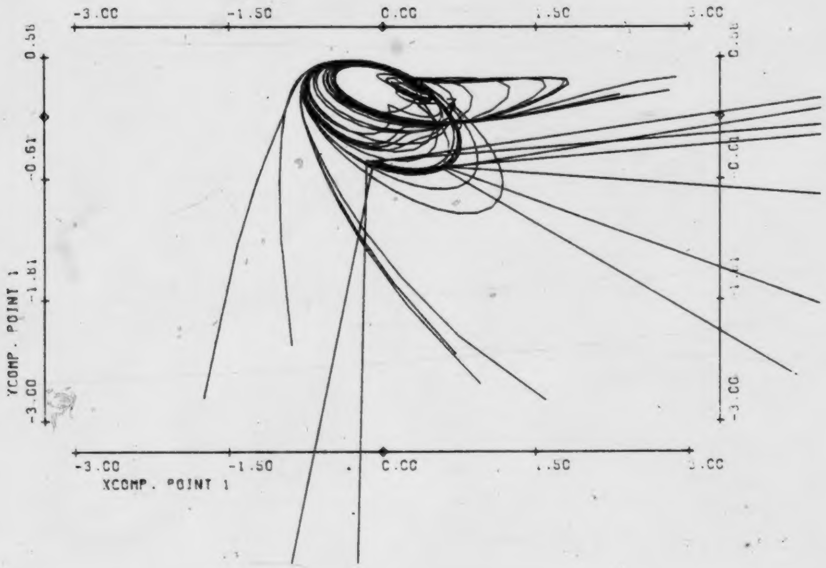
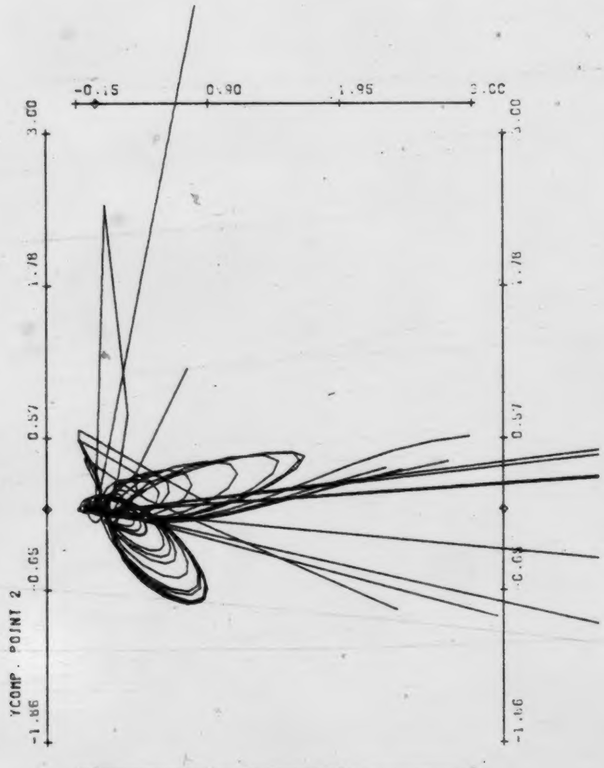
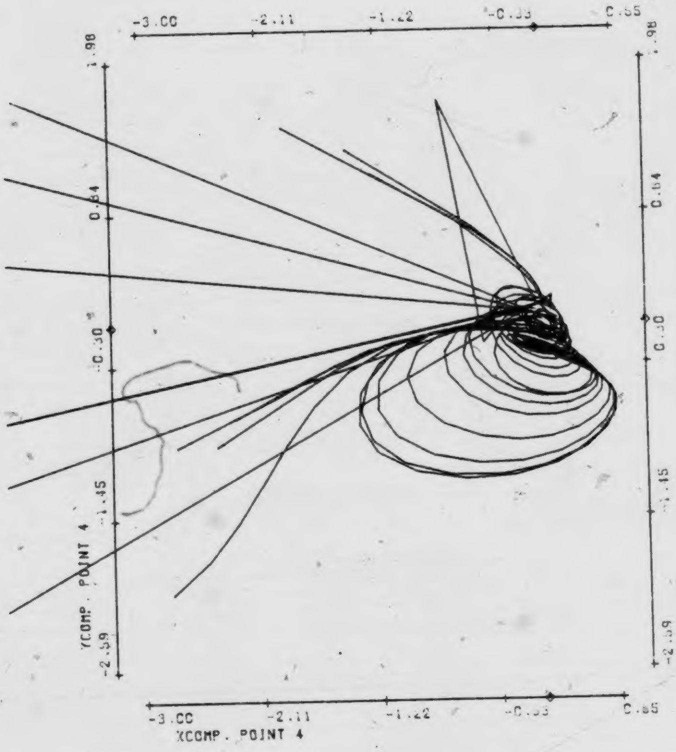

-1.02
-0.73
-0.45
-0.16
0.12
1.11
0.82
0.52
0.22
-0.07
ZCOMP. POINT 4
YCOMP. POINT 4

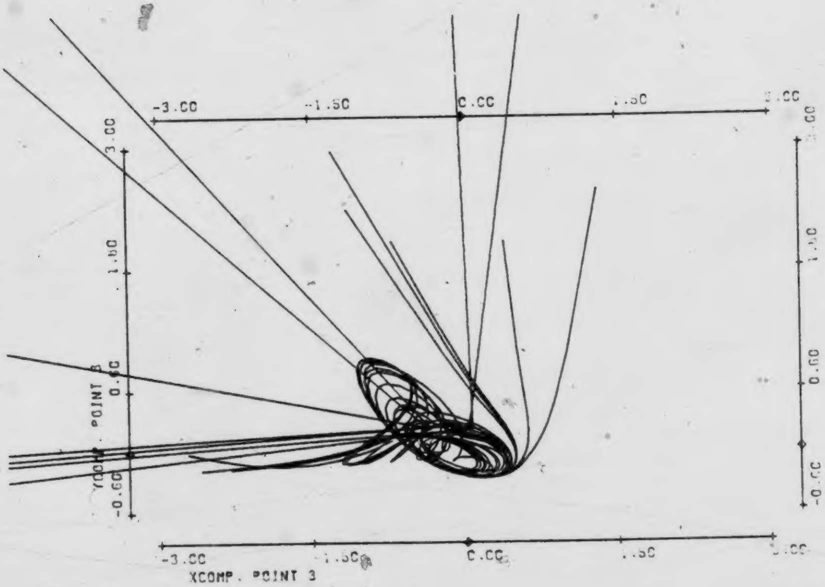
FIGURE T9

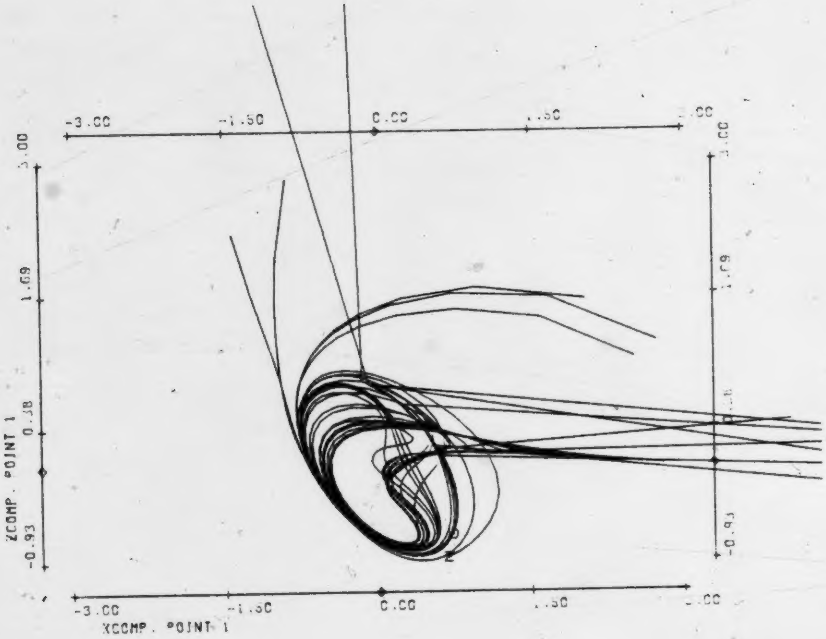
Bifurcation parameter η	.47
No. of timesteps	1000
Stepsize	.04
Init. condition for $\vec{V}_2$	.01(X-comp.Pt.1) rest = 0
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.53
Mean velocity:	
Point 1	1.71
2	1.76
3	2.78
4	1.83

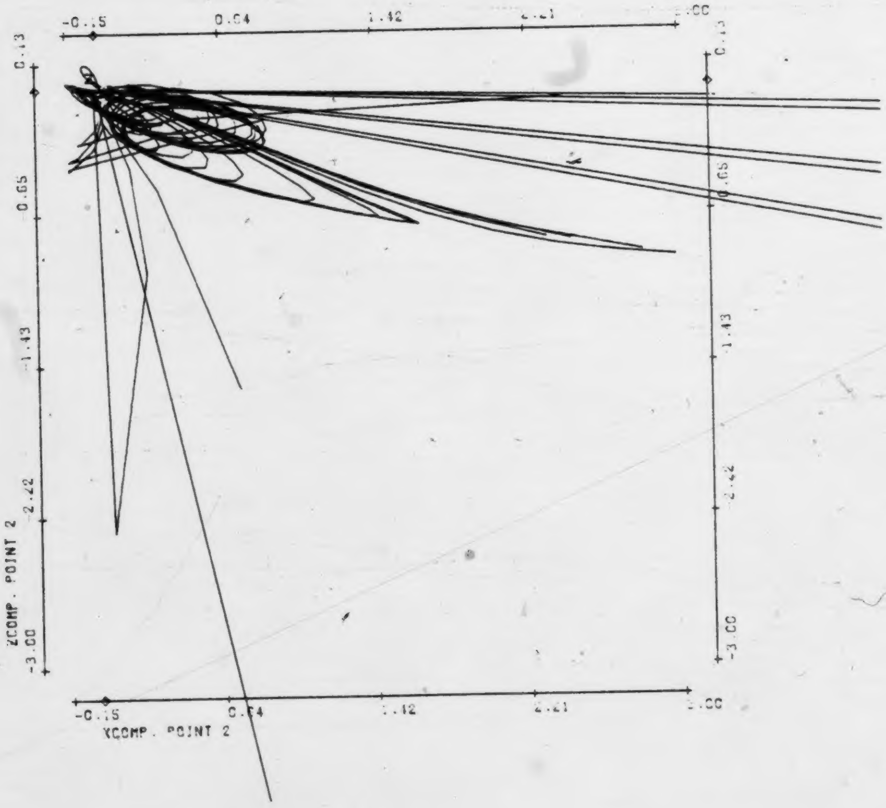

-3.00
-1.50
0.00
1.50
3.00
0.58
-0.61
-1.81
-3.00
YCOMP. POINT 1
XCOMP. POINT 1

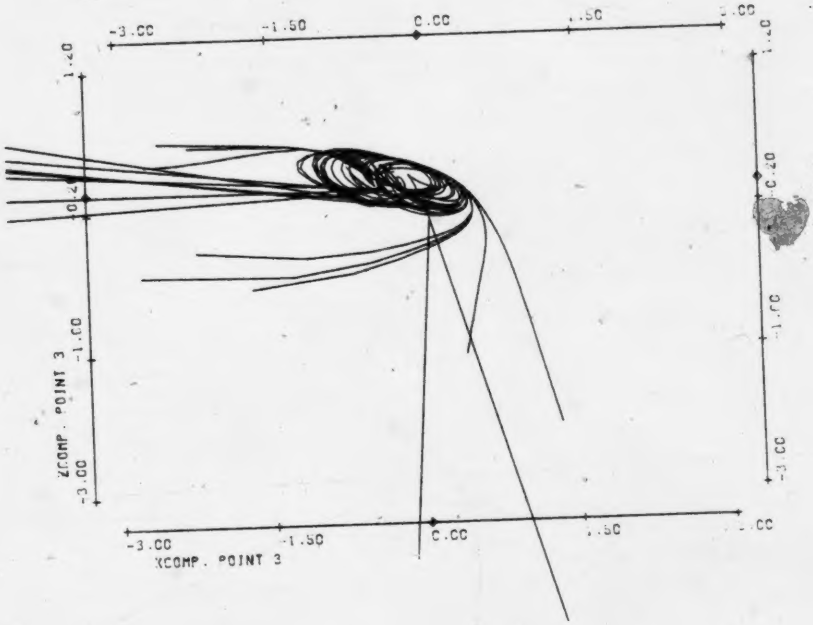

-0.15
0.90
1.95
3.00
3.00
1.78
0.57
-0.65
-1.86
YCOMP. POINT 2
XCOMP. POINT 2

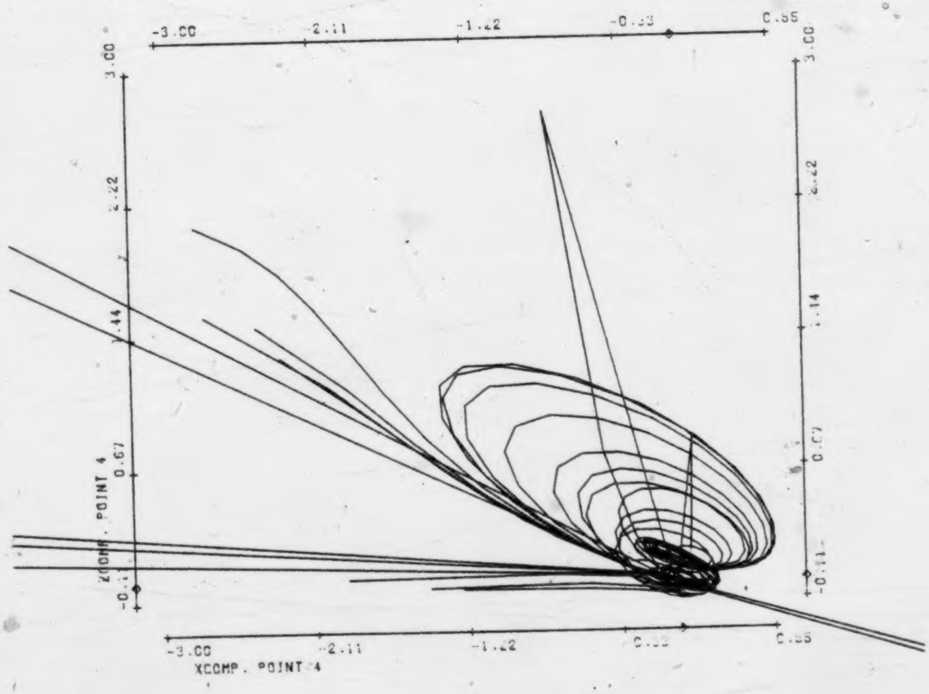

-3.00
-2.11
-1.22
-0.33
0.55
1.98
0.84
-0.30
-1.45
-2.59
YCOMP. POINT 4
XCOMP. POINT 4


-3.00
-1.50
0.00
1.50
3.00
3.00
1.50
0.00
-0.60
YCOMP. POINT 3
XCOMP. POINT 3

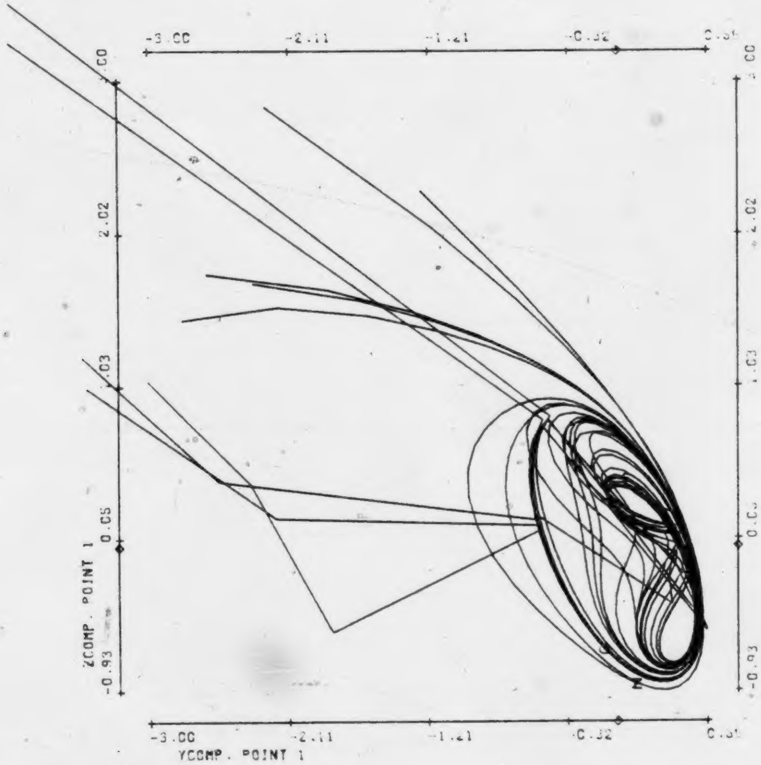

-3.00
-1.50
0.00
1.50
3.00
3.00
1.69
0.38
-0.93
ZCOMP. POINT 1
XCOMP. POINT 1

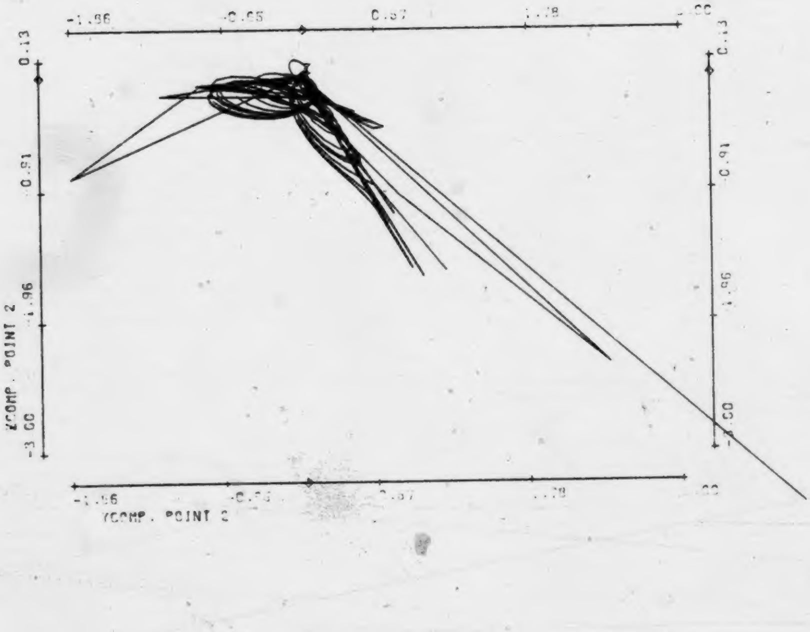

-0.15
0.64
1.42
2.21
3.00
0.13
-0.65
-1.43
-2.22
-3.00
ZCOMP. POINT 2
XCOMP. POINT 2

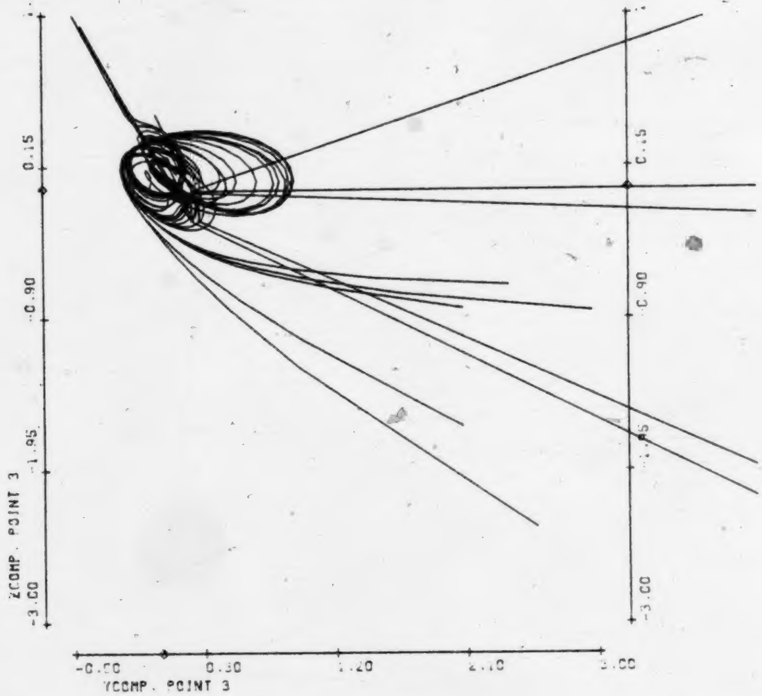

-3.00
-1.50
0.00
1.50
3.00
1.20
0.20
-1.00
-3.00
ZCOMP. POINT 3
XCOMP. POINT 3


-3.00
-2.11
-1.22
-0.33
0.55
3.00
2.22
1.44
0.67
-0.11
ZCOMP. POINT 4
XCOMP. POINT 4

T9(5)


-3.00
-2.11
-1.21
-0.32
3.00
2.02
1.03
0.05
-0.93
ZCOMP. POINT 1
YCOMP. POINT 1


-1.96
-0.65
0.57
1.78
3.00
0.13
-0.91
-1.96
-3.00
ZCOMP. POINT 2
YCOMP. POINT 2


-0.60
0.30
1.20
2.10
3.00
1.20
0.15
-0.90
-1.95
-3.00
ZCOMP. POINT 3
YCOMP. POINT 3

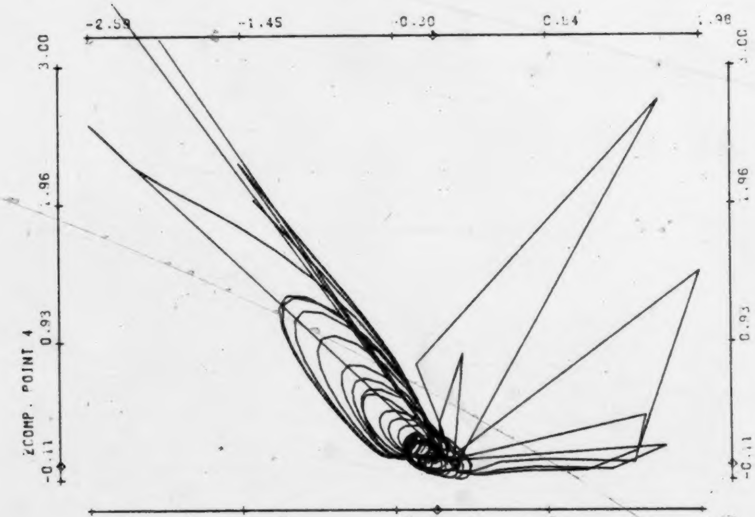
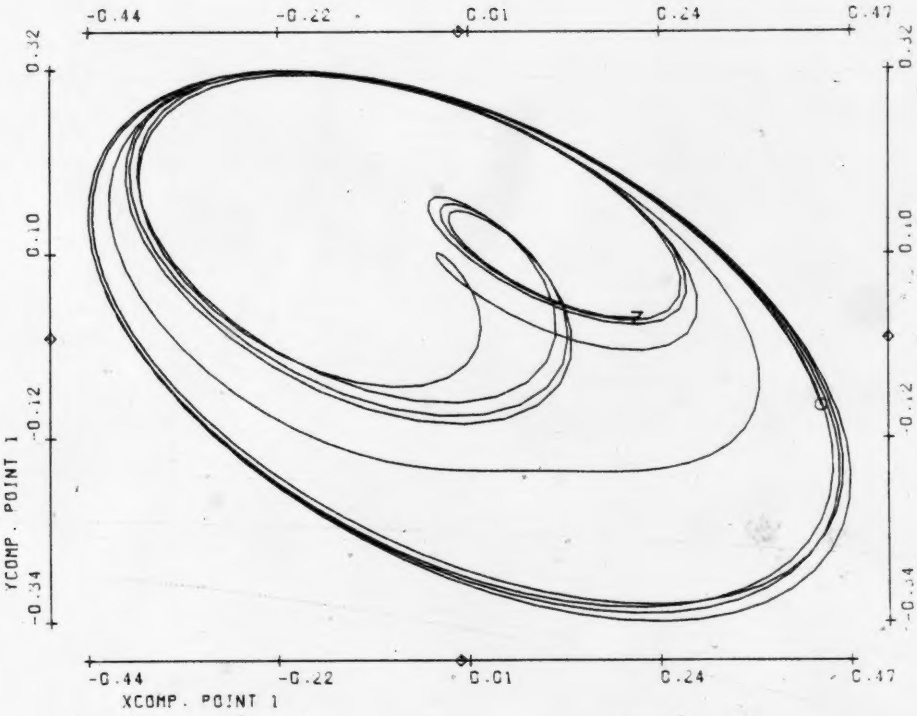
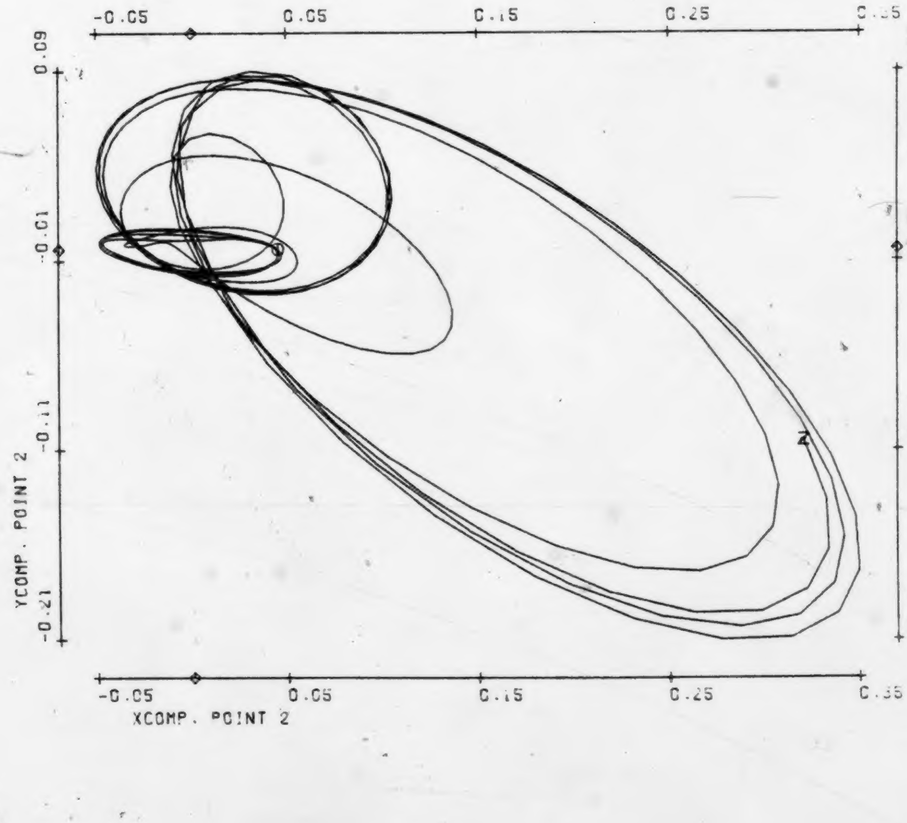

-2.59
-1.45
-0.30
0.84
1.98
3.00
1.96
0.93
-0.11
ZCOMP. POINT 4
YCOMP. POINT 4

FIGURE T10

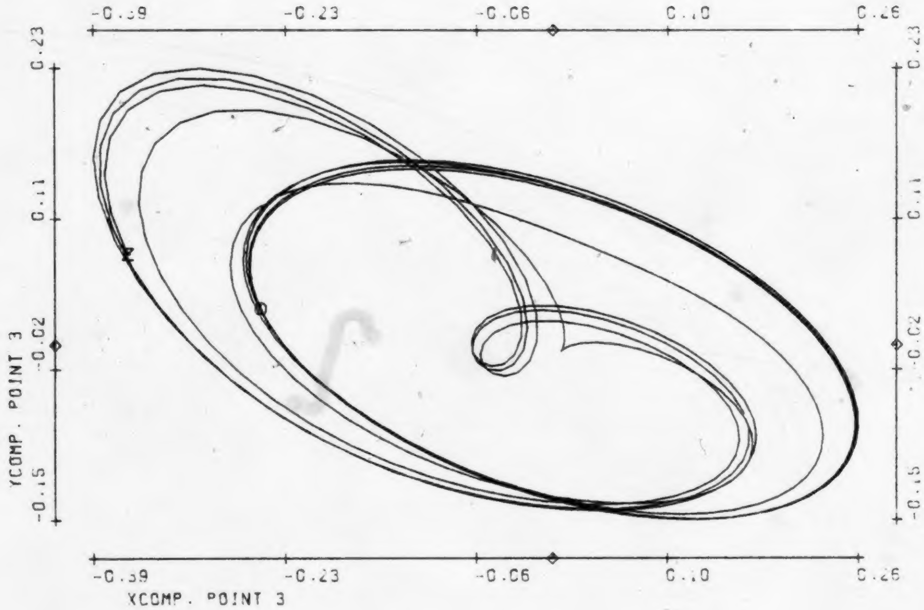
Bifurcation parameter η	.32
No. of timesteps	500
Stepsize	.03
Init. condition for $\vec{V}_2$	.001 (X-comp.Pt.1) rest = 0
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.66
Mean velocity:	
Point 1	1.95
2	1.64
3	2.65
4	1.60

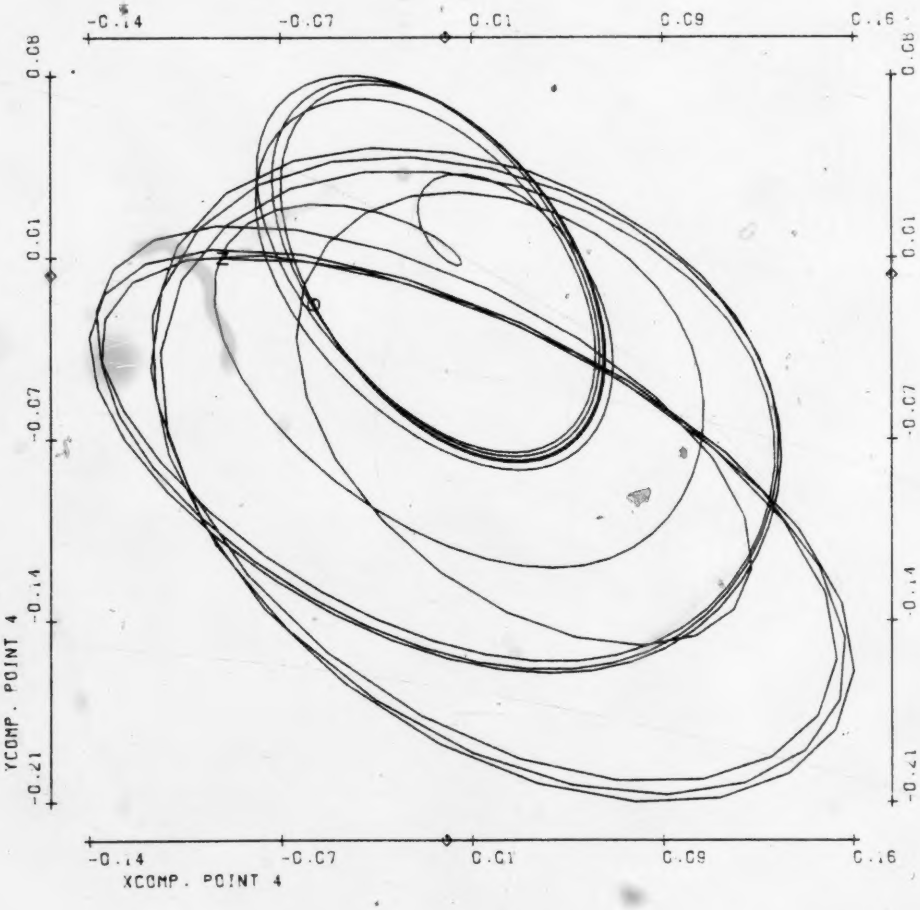
T10(1)


-0.44
-0.22
0.01
0.24
0.47
0.32
0.10
-0.12
-0.34
YCOMP. POINT 1
XCOMP. POINT 1
Z

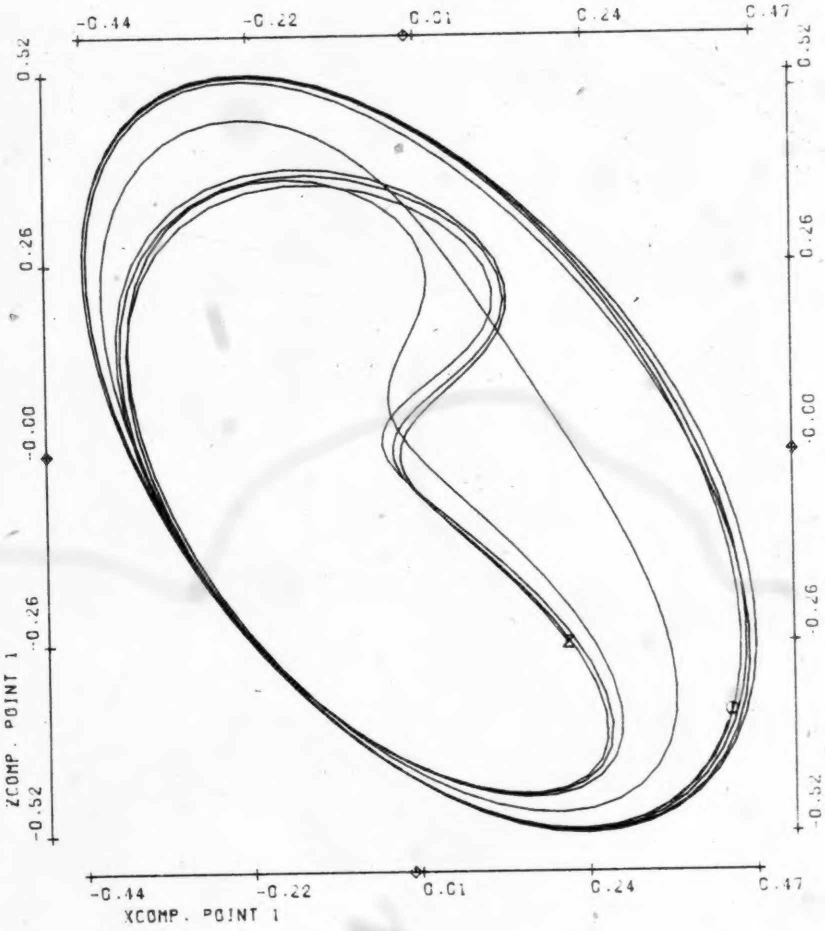

-0.05
0.05
0.15
0.25
0.35
0.09
-0.01
-0.11
-0.21
YCOMP. POINT 2
XCOMP. POINT 2
Z

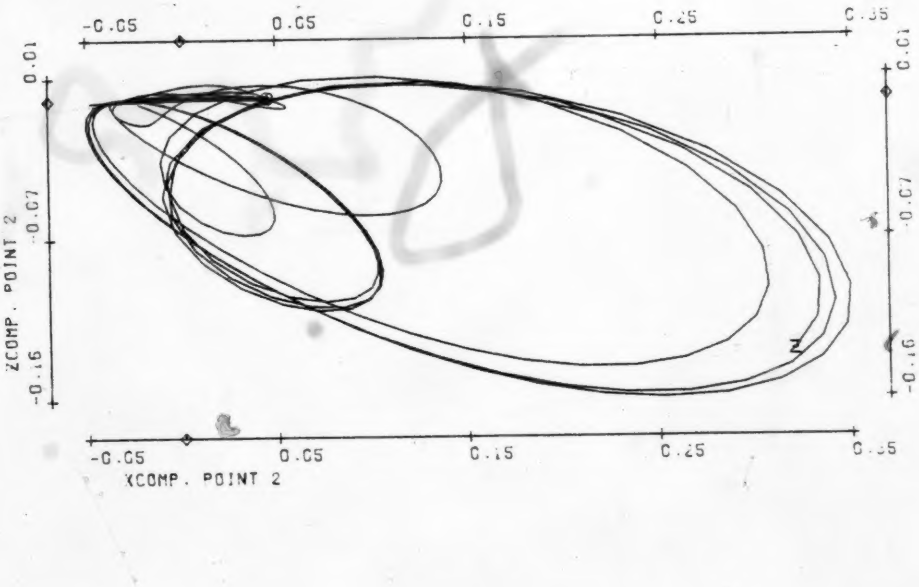
T10(2)


-0.39
-0.23
-0.06
0.10
0.26
0.23
0.11
-0.02
-0.15
Z
O
YCOMP. POINT 3
XCOMP. POINT 3

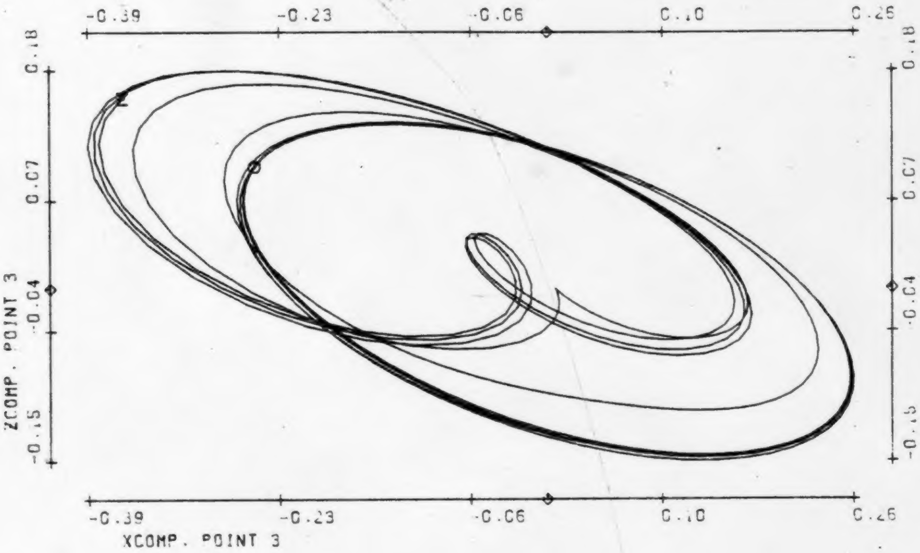

-0.14
-0.07
0.01
0.09
0.16
0.08
0.01
-0.07
-0.14
-0.21
Z
O
YCOMP. POINT 4
XCOMP. POINT 4

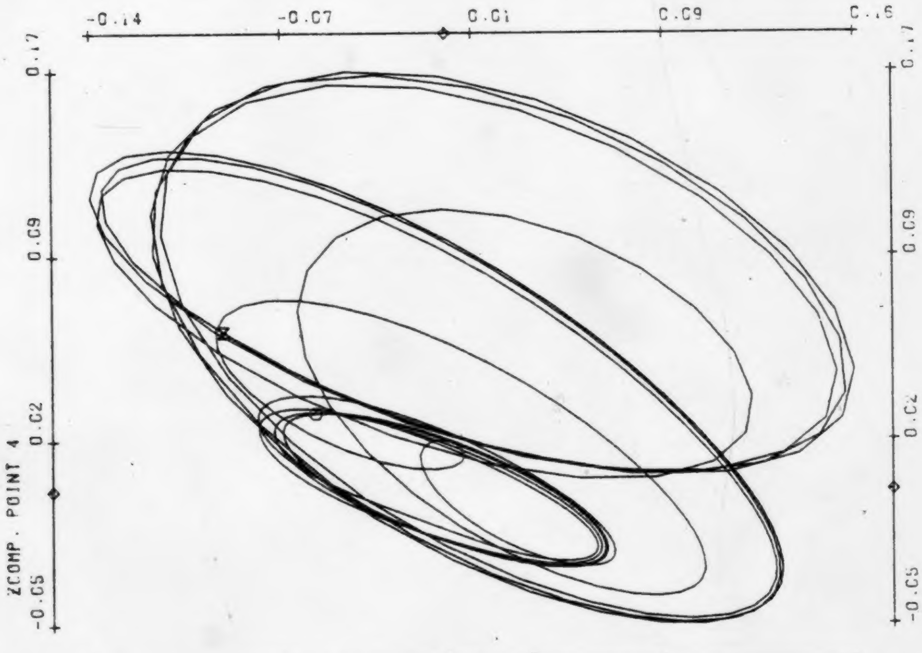
T10(3)


-0.44
-0.22
0.01
0.24
0.47
0.52
0.26
-0.00
-0.26
-0.52
ZCOMP. POINT 1
XCOMP. POINT 1

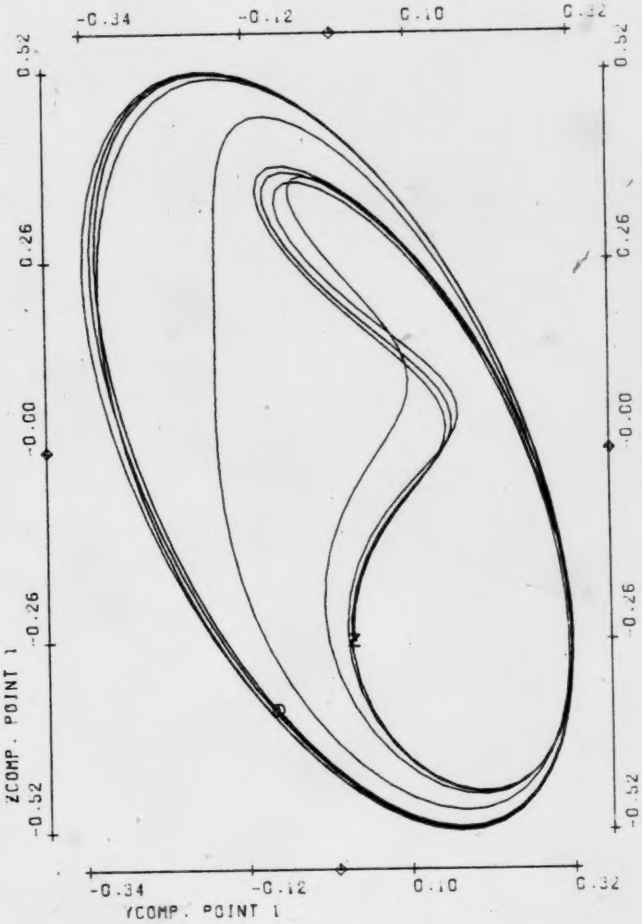

-0.05
0.05
0.15
0.25
0.35
0.01
-0.07
-0.16
ZCOMP. POINT 2
XCOMP. POINT 2

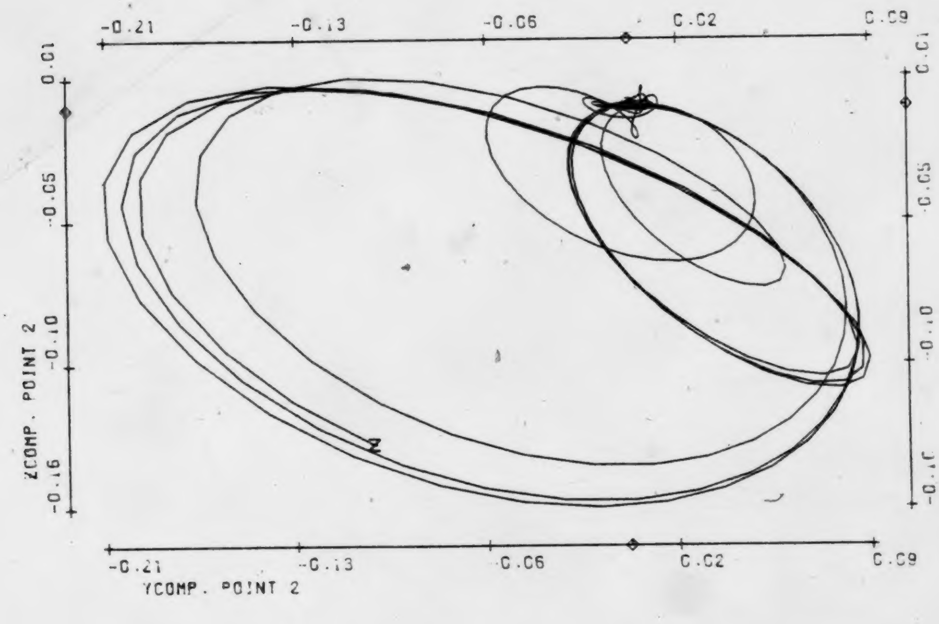
T10(4)

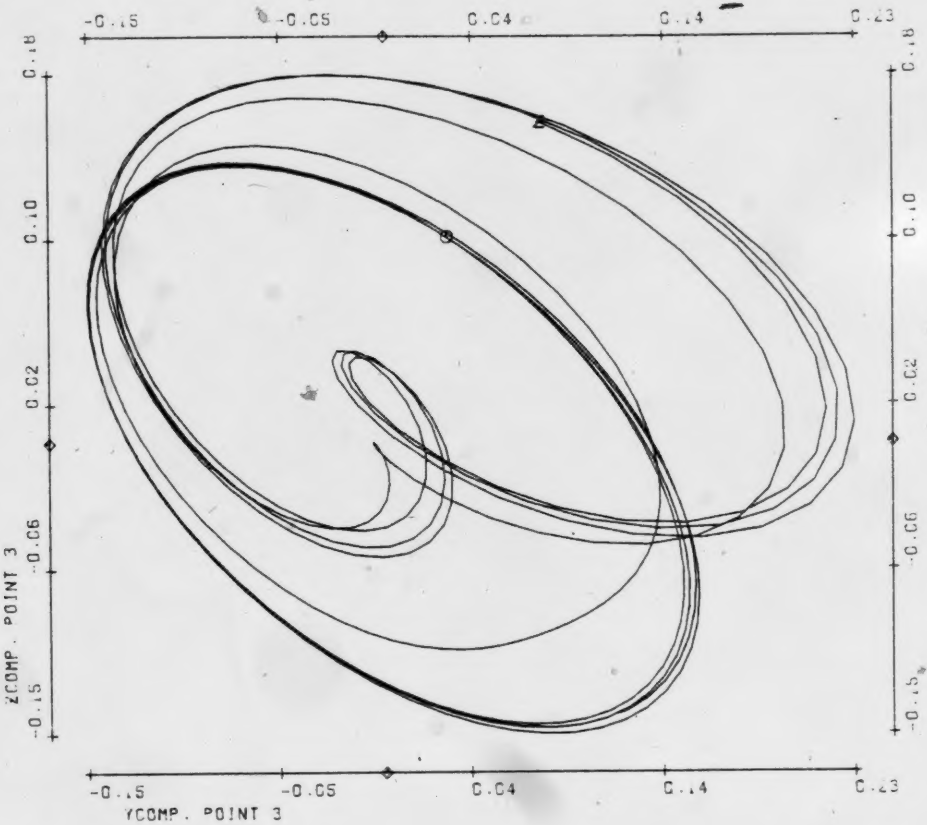

-0.39
-0.23
-0.06
0.10
0.26
0.18
0.07
-0.04
-0.15
ZCOMP. POINT 3
XCOMP. POINT 3


-0.14
-0.07
0.01
0.09
0.16
0.17
0.09
0.02
-0.05
ZCOMP. POINT 4
XCOMP. POINT 4

T10(5)


-0.34
-0.12
0.10
0.32
0.52
0.26
-0.00
-0.26
-0.52
Z
ZCOMP. POINT 1
YCOMP. POINT 1


-0.21
-0.13
-0.06
0.02
0.09
0.01
-0.05
-0.10
-0.16
Z
ZCOMP. POINT 2
YCOMP. POINT 2


-0.15
-0.05
0.04
0.14
0.23
0.18
0.10
0.02
-0.06
-0.15
ZCOMP. POINT 3
YCOMP. POINT 3

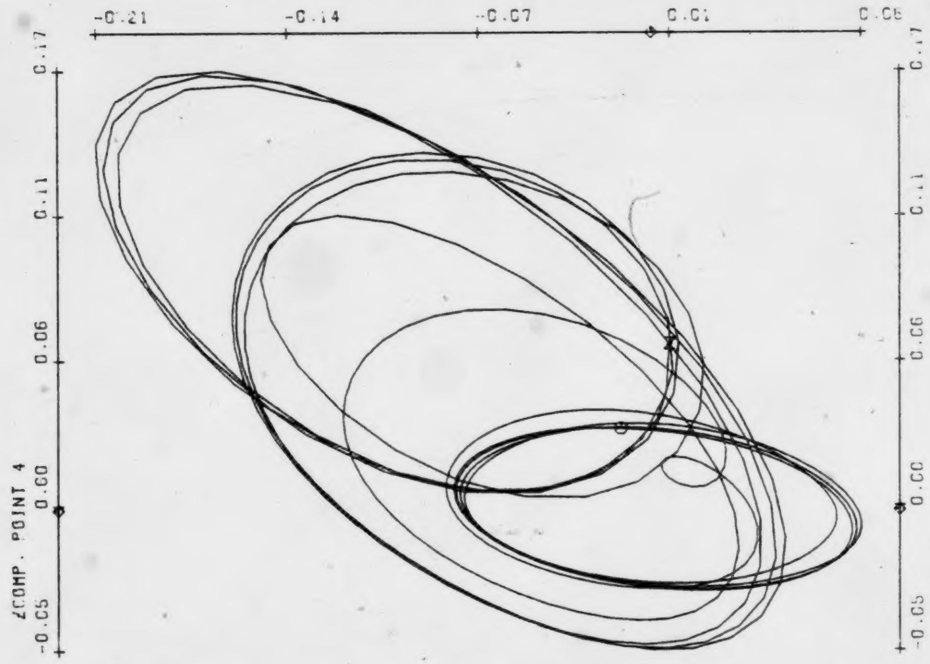
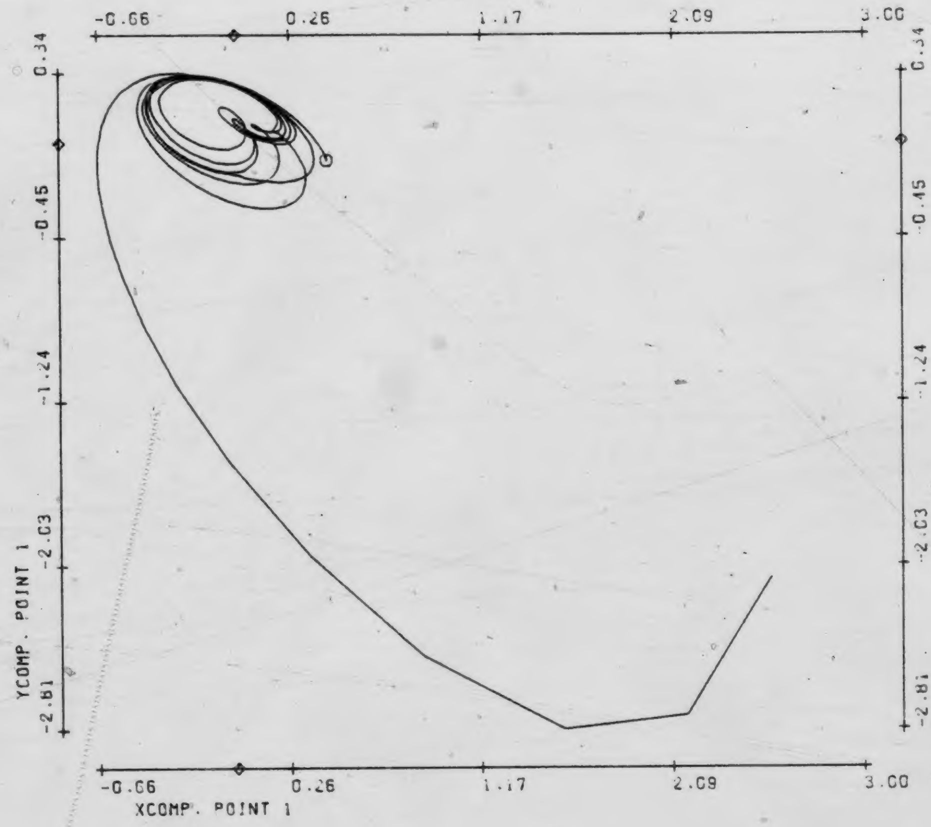
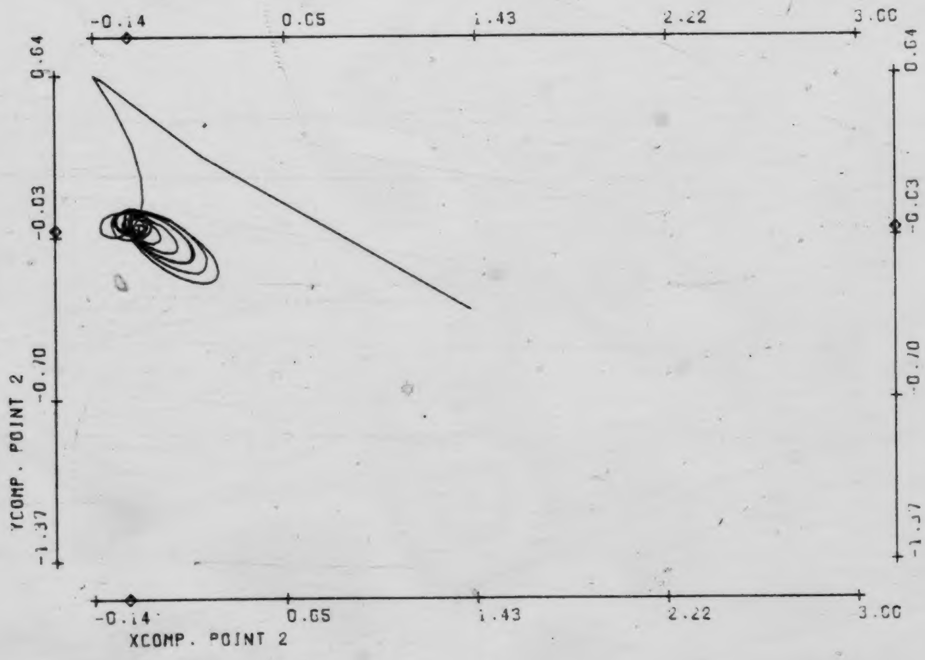

-0.21
-0.14
-0.07
0.01
0.08
0.17
0.11
0.06
0.00
-0.05
ZCOMP. POINT 4
YCOMP. POINT 4

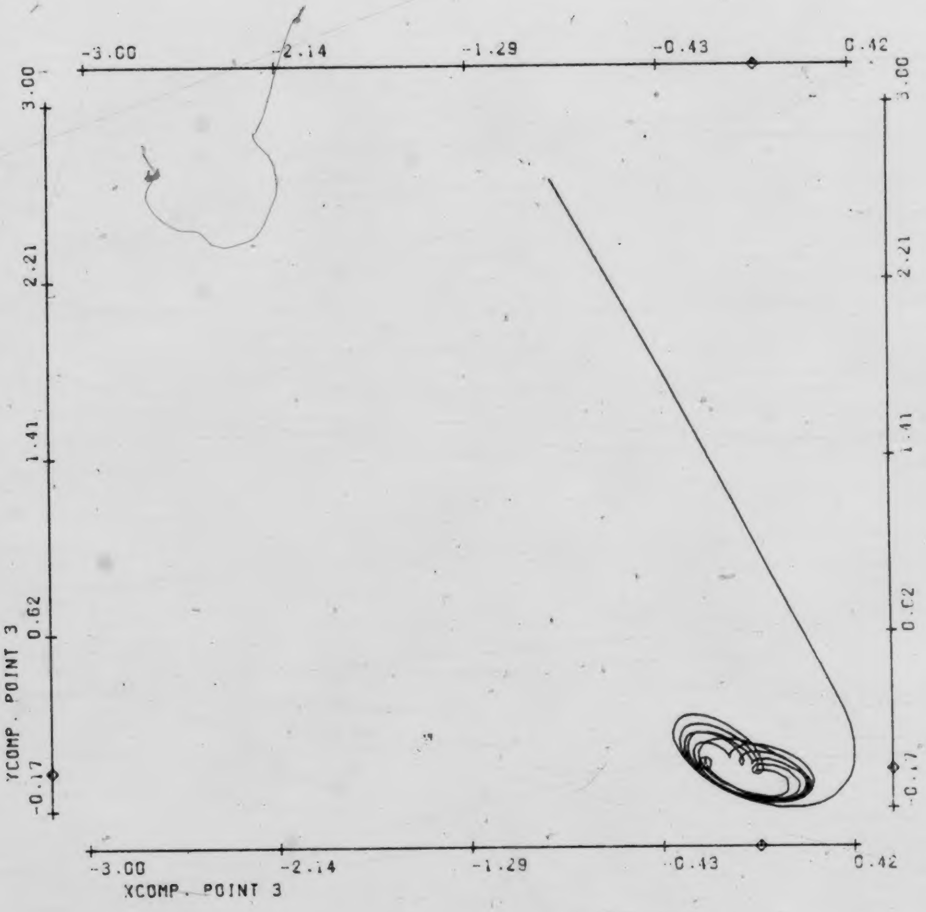
FIGURE T11

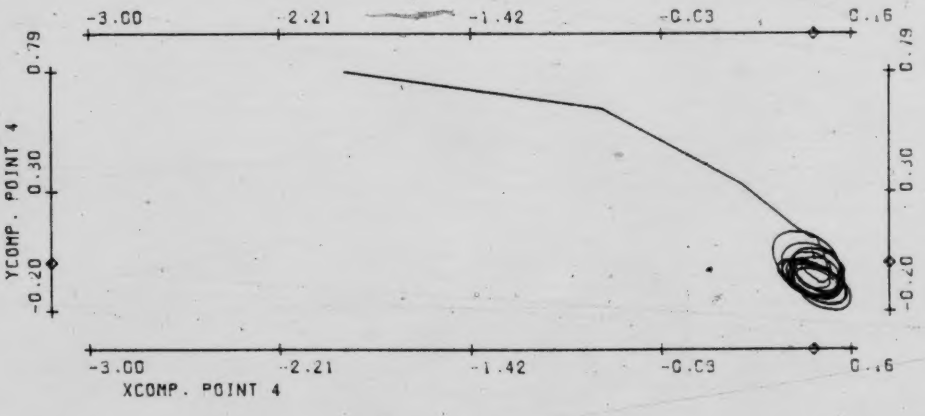
Bifurcation parameter η	.32
No. of timesteps	379
Stepsize	.03
Init. condition for $\vec{V}_2$	0 (all)
Average size of spatial random perturbations	.005
Average no. of timesteps between random perturbations	2
Period of $\vec{V}_1$	1.66
Mean velocity:	
Point 1	1.95
2	1.64
3	2.65
4	1.60

T11(1)

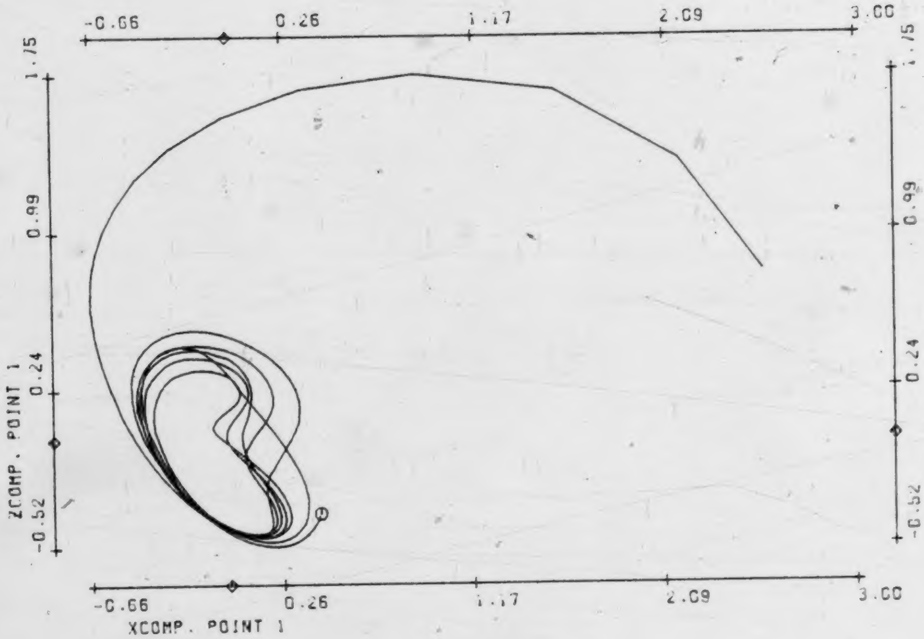

-0.66
0.26
1.17
2.09
3.00
0.34
-0.45
-1.24
-2.03
-2.81
XCOMP. POINT 1
YCOMP. POINT 1

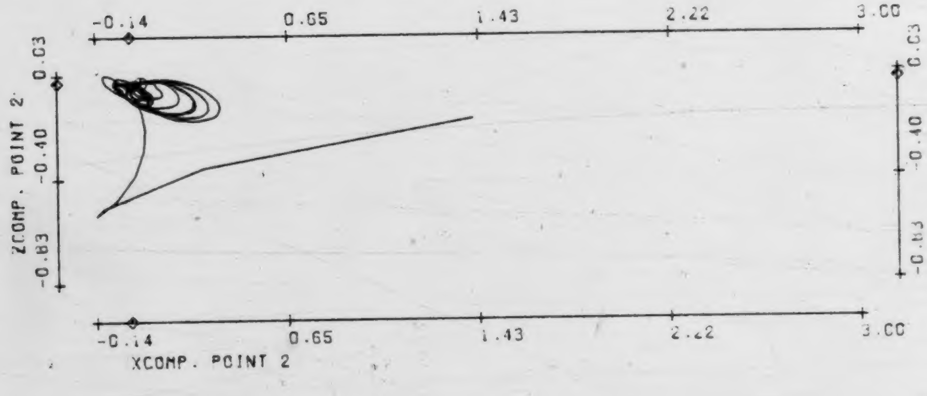

-0.14
0.65
1.43
2.22
3.00
0.64
-0.03
-0.70
-1.37
XCOMP. POINT 2
YCOMP. POINT 2


-3.00
-2.14
-1.29
-0.43
0.42
3.00
2.21
1.41
0.62
-0.17
YCOMP. POINT 3
XCOMP. POINT 3

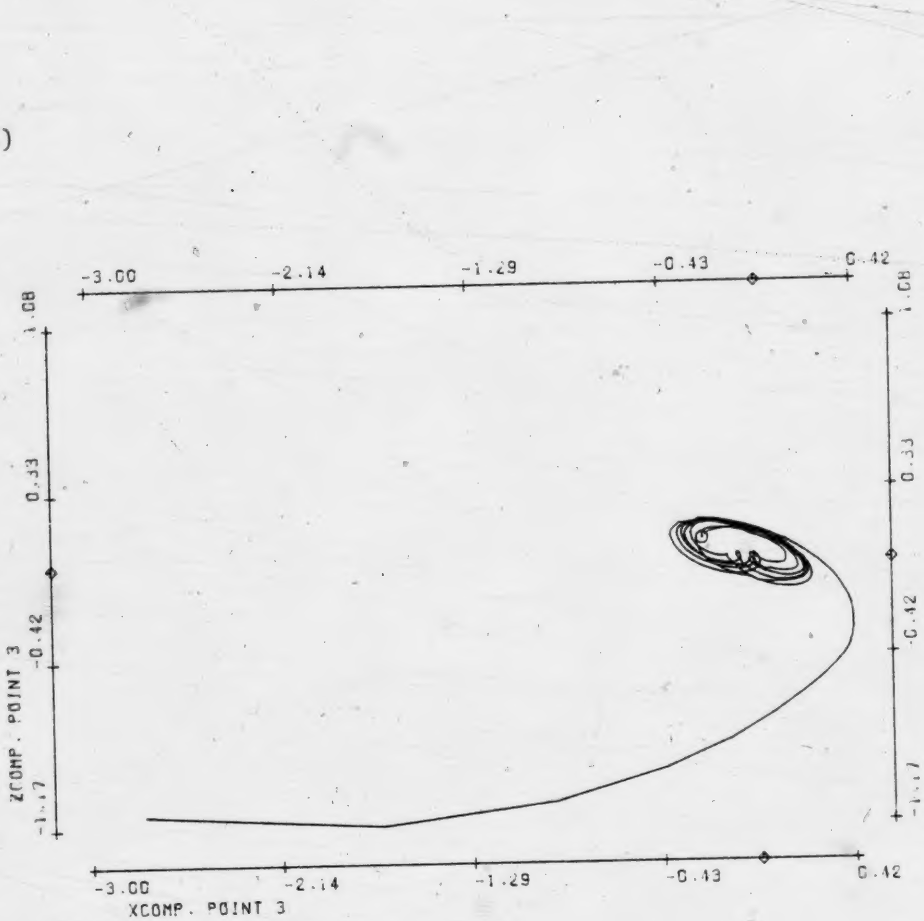

-3.00
-2.21
-1.42
-0.63
0.16
0.79
0.30
-0.20
YCOMP. POINT 4
XCOMP. POINT 4

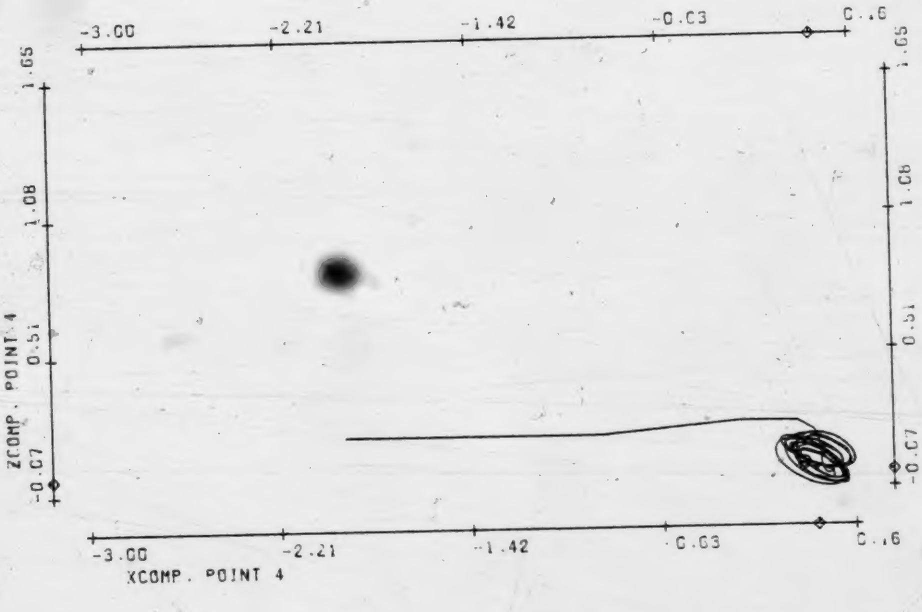
T11(3)


-0.66
0.26
1.17
2.09
3.00
1.75
0.99
0.24
-0.52
ZCOMP. POINT 1
XCOMP. POINT 1

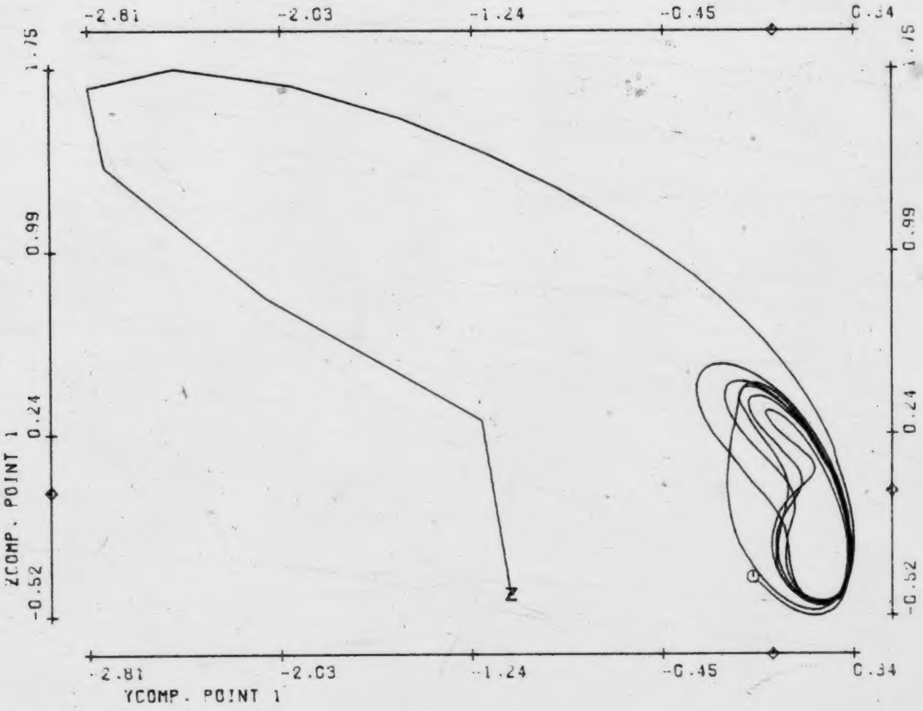

-0.14
0.65
1.43
2.22
3.00
0.03
-0.40
-0.83
ZCOMP. POINT 2
XCOMP. POINT 2

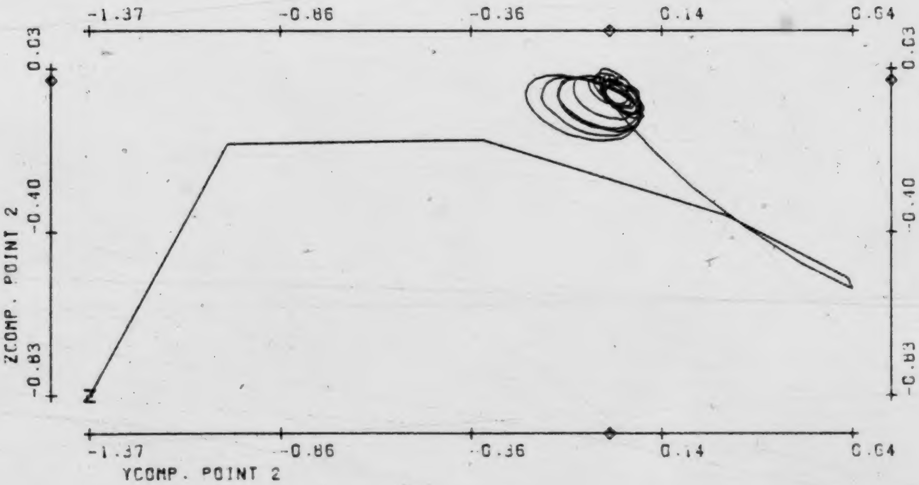
T11(4)


-3.00
-2.14
-1.29
-0.43
0.42
1.08
0.33
-0.42
-1.17
ZCOMP. POINT 3
XCOMP. POINT 3

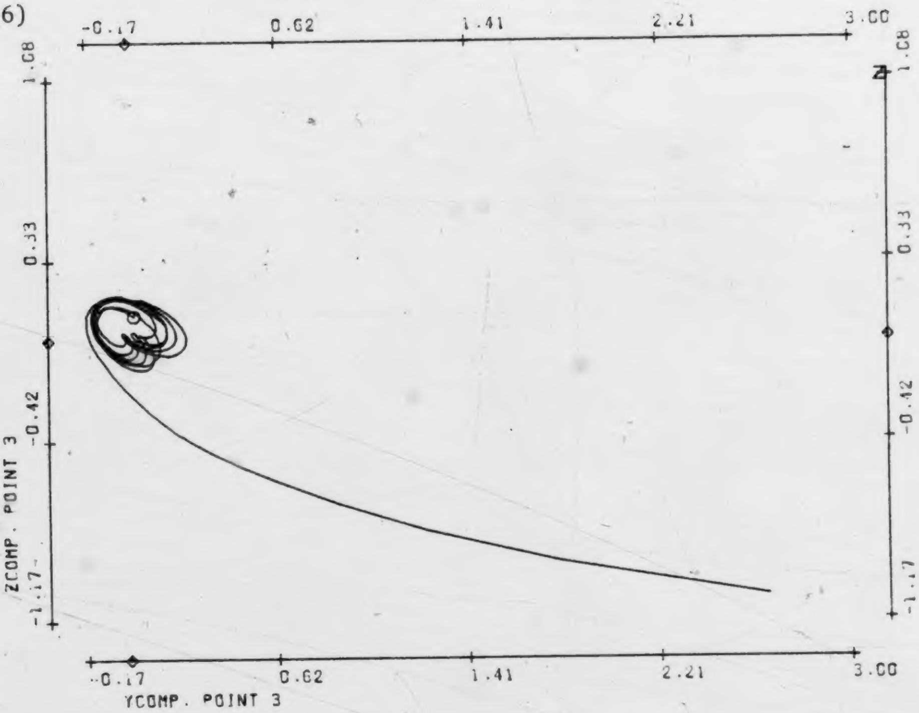

-3.00
-2.21
-1.42
-0.63
0.16
1.65
1.08
0.51
-0.07
ZCOMP. POINT 4
XCOMP. POINT 4

T11(5)


-2.81
-2.03
-1.24
-0.45
0.34
1.75
0.99
0.24
-0.52
ZCOMP. POINT 1
YCOMP. POINT 1
Z


-1.37
-0.86
-0.36
0.14
0.64
0.03
-0.40
-0.83
ZCOMP. POINT 2
YCOMP. POINT 2
Z

T11(6)


-0.17
0.62
1.41
2.21
3.00
1.08
0.33
-0.42
-1.17
Z
ZCOMP. POINT 3
YCOMP. POINT 3

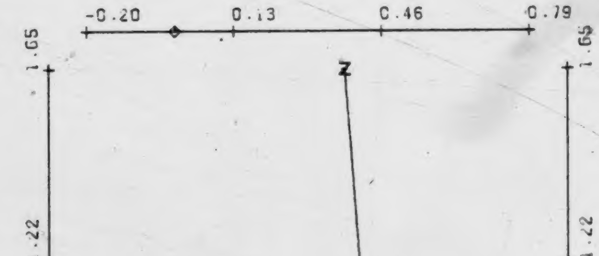
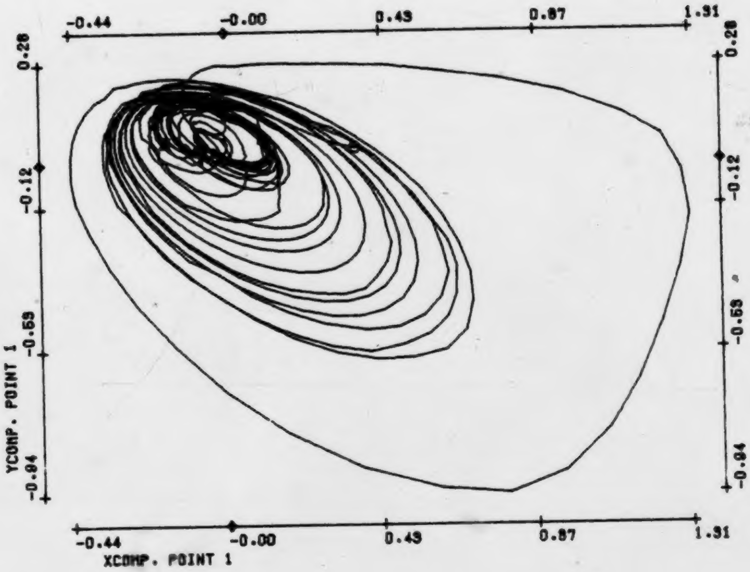
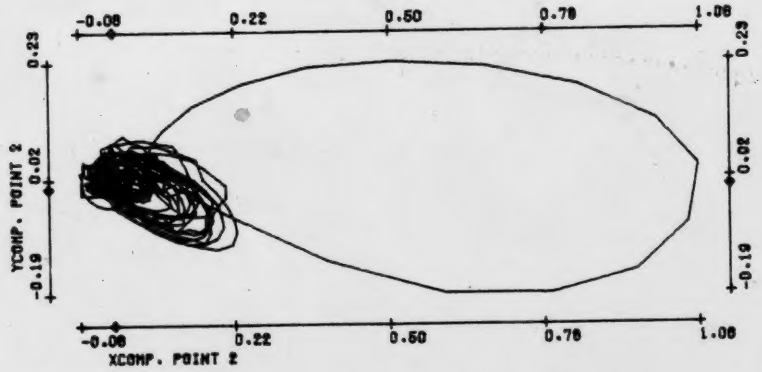

-0.20
0.13
0.46
0.79
1.65
1.22
0.79
0.36
-0.07
Z
ZCOMP. POINT 4
YCOMP. POINT 4

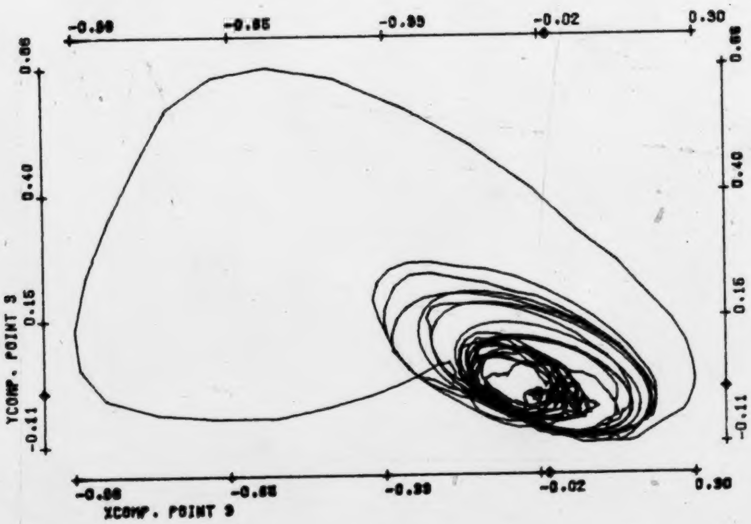
FIGURE T12

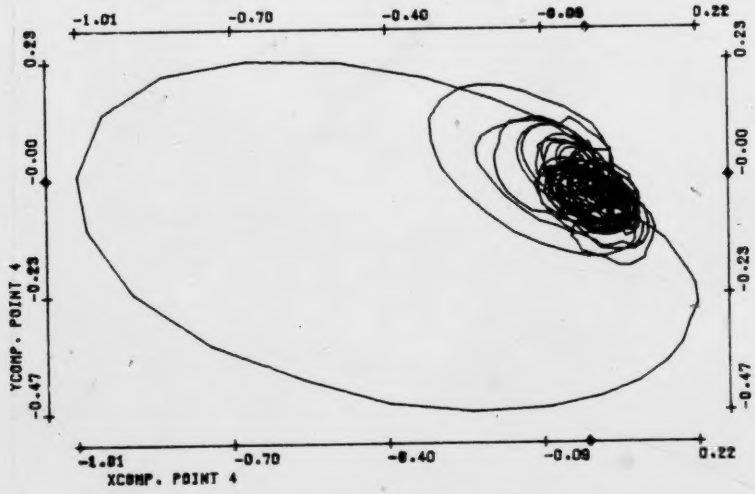
Bifurcation parameter η	.27
No. of timesteps	1000
Stepsize	.04
Init. condition for $\vec{V}_2$	.1 (all)
Average size of spatial random perturbations	.01
Average no. of timesteps between random perturbations	1.5
Period of $\vec{V}_1$	1.7
Mean velocity:	
Point 1	1.68
2	1.56
3	2.43
4	1.56

(First 233 timesteps are unperturbed.)

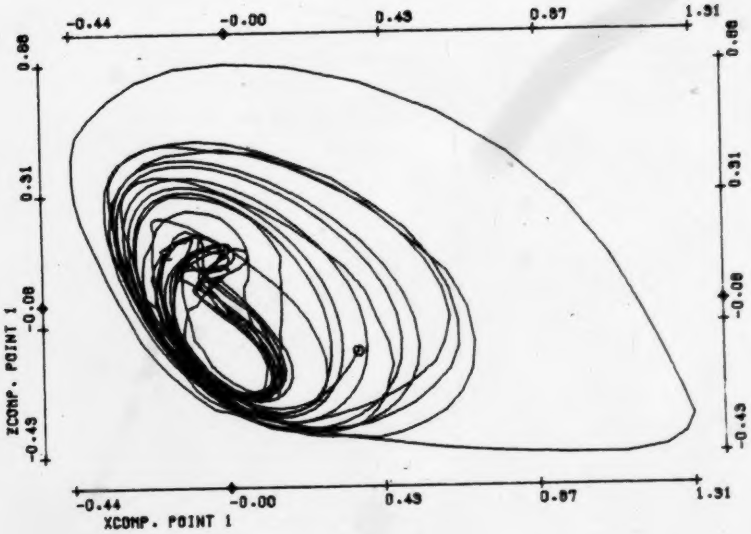

-0.44
-0.00
0.43
0.87
1.31
0.28
-0.12
-0.53
-0.94
YCOMP. POINT 1
XCOMP. POINT 1

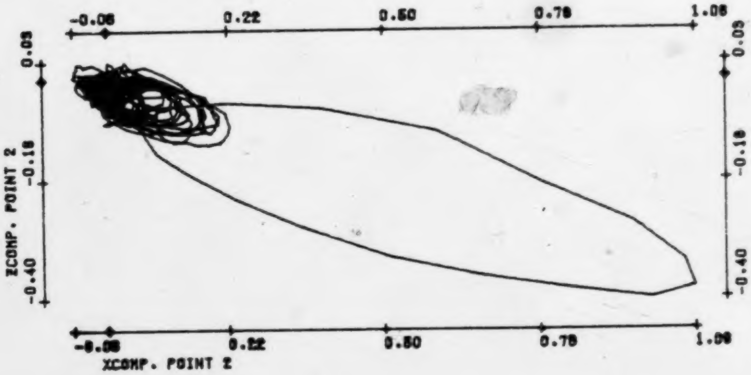

-0.08
0.22
0.50
0.78
1.08
0.23
0.02
-0.19
YCOMP. POINT 2
XCOMP. POINT 2

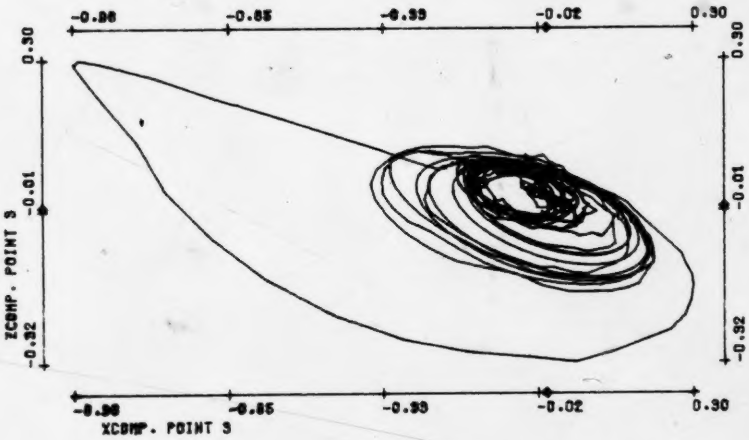

-0.98
-0.66
-0.34
-0.02
0.30
0.66
0.40
0.15
-0.11
YCOMP. POINT 3
0.66
0.40
0.15
-0.11
-0.98
-0.66
-0.34
-0.02
0.30
XCOMP. POINT 3

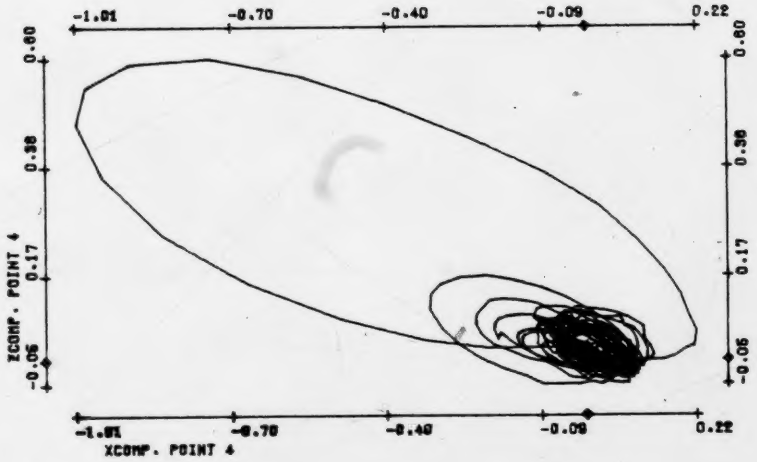

-1.01
-0.70
-0.40
-0.09
0.22
0.23
-0.00
-0.23
-0.47
YCOMP. POINT 4
0.23
-0.00
-0.23
-0.47
-1.01
-0.70
-0.40
-0.09
0.22
XCOMP. POINT 4

T12(3)

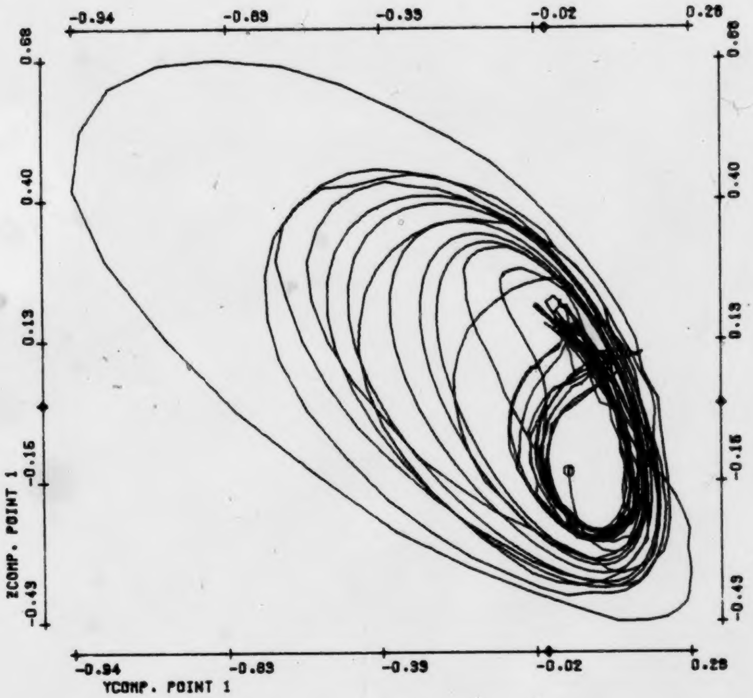

-0.44
-0.00
0.43
0.87
1.31
0.68
0.31
-0.06
-0.43
ZCOMP. POINT 1
XCOMP. POINT 1

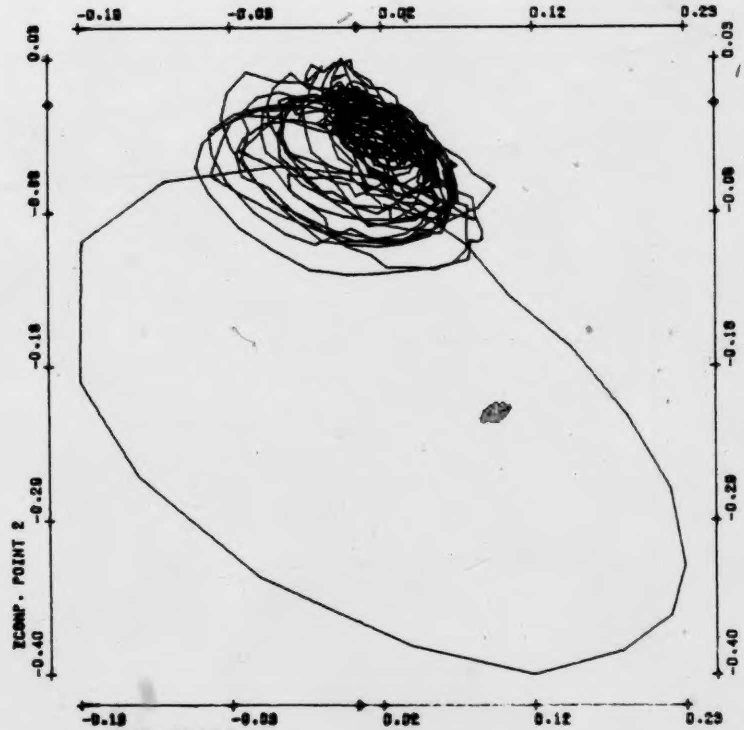

-0.06
0.22
0.50
0.78
1.06
0.05
-0.18
-0.40
ZCOMP. POINT 2
XCOMP. POINT 2


-0.96
-0.65
-0.33
-0.02
0.30
0.30
-0.01
-0.32
ZCOMP. POINT 3
XCOMP. POINT 3

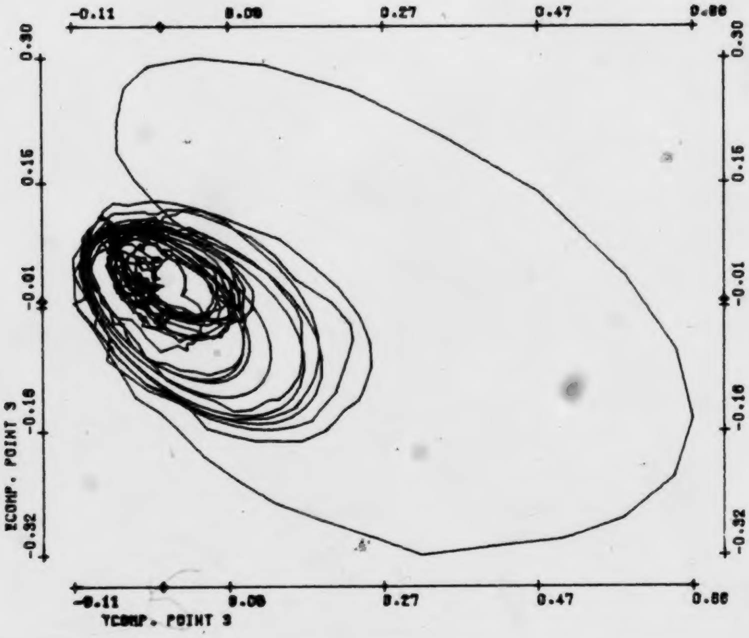

-1.01
-0.70
-0.40
-0.09
0.22
0.60
0.38
0.17
-0.06
ZCOMP. POINT 4
XCOMP. POINT 4

T12(5)


-0.94
-0.63
-0.33
-0.02
0.28
0.68
0.40
0.13
-0.16
-0.43
ZCOMP. POINT 1
YCOMP. POINT 1


-0.19
-0.08
0.02
0.12
0.23
0.03
-0.08
-0.18
-0.29
-0.40
ZCOMP. POINT 2
YCOMP. POINT 2

T12(6)


-0.11
0.08
0.27
0.47
0.66
0.30
0.15
-0.01
-0.16
-0.32
ZCOMP. POINT 3
YCOMP. POINT 3

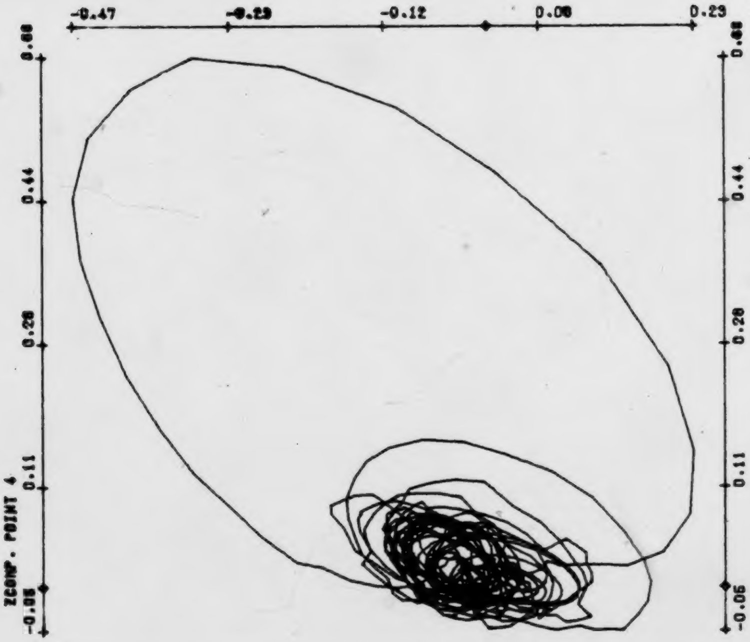
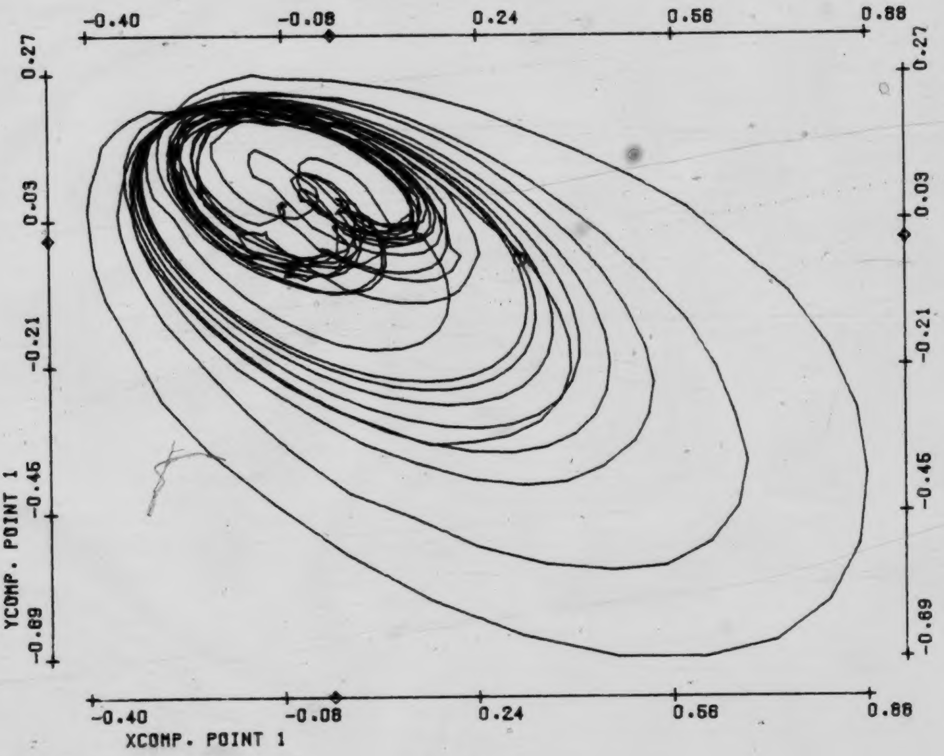

-0.47
-0.29
-0.12
0.06
0.23
0.60
0.44
0.28
0.11
-0.06
ZCOMP. POINT 4
YCOMP. POINT 4

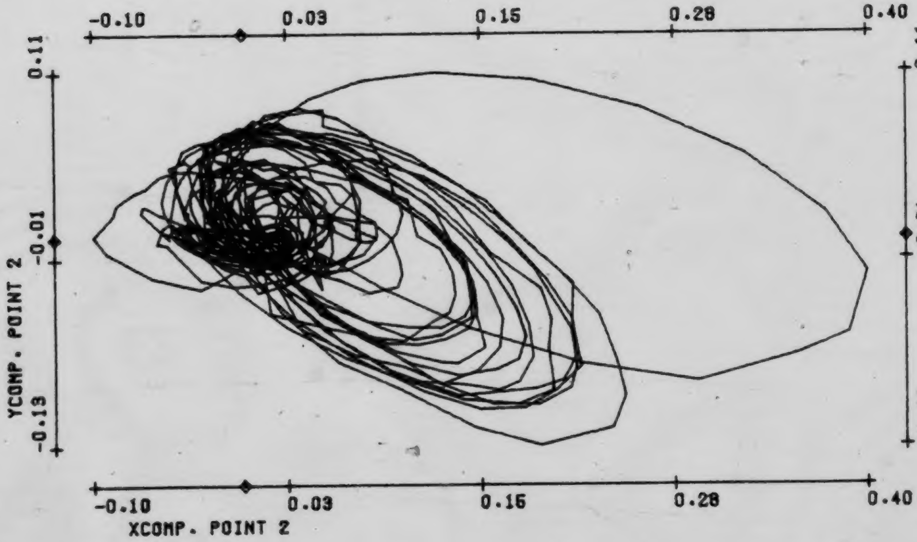
FIGURE T13

Bifurcation parameter η	.28
No. of timesteps	1000
Stepsize	.04
Init. condition for $\vec{V}_2$	.01 (all)
Average size of spatial random perturbations	.01
Average no. of timesteps between random perturbations	2
Period of $\vec{V}_1$	1.69
Mean velocity:	
Point 1	1.68
2	1.55
3	2.42
4	1.56

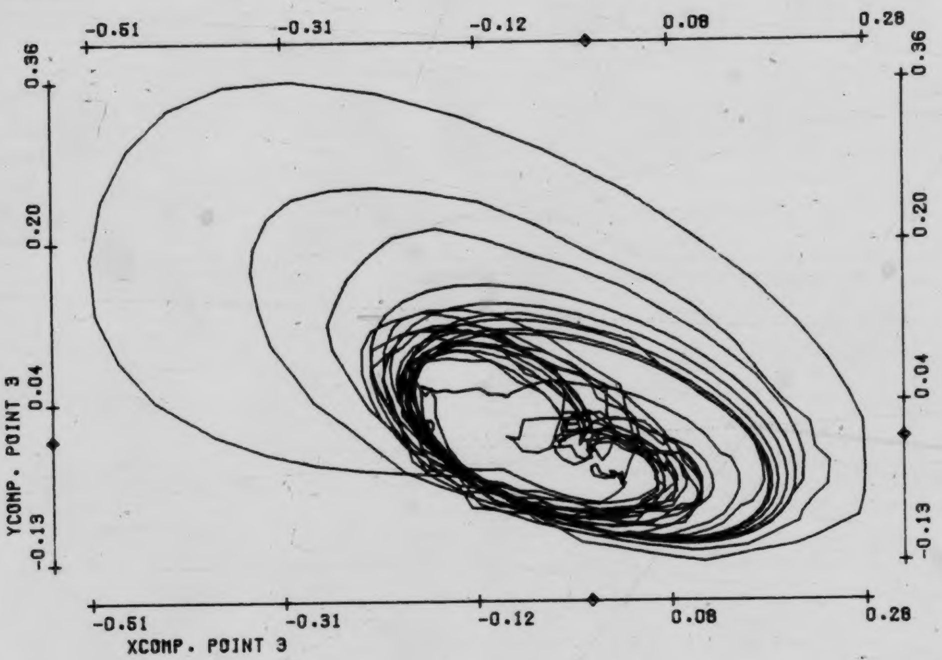
(First 233 timesteps are unperturbed.)

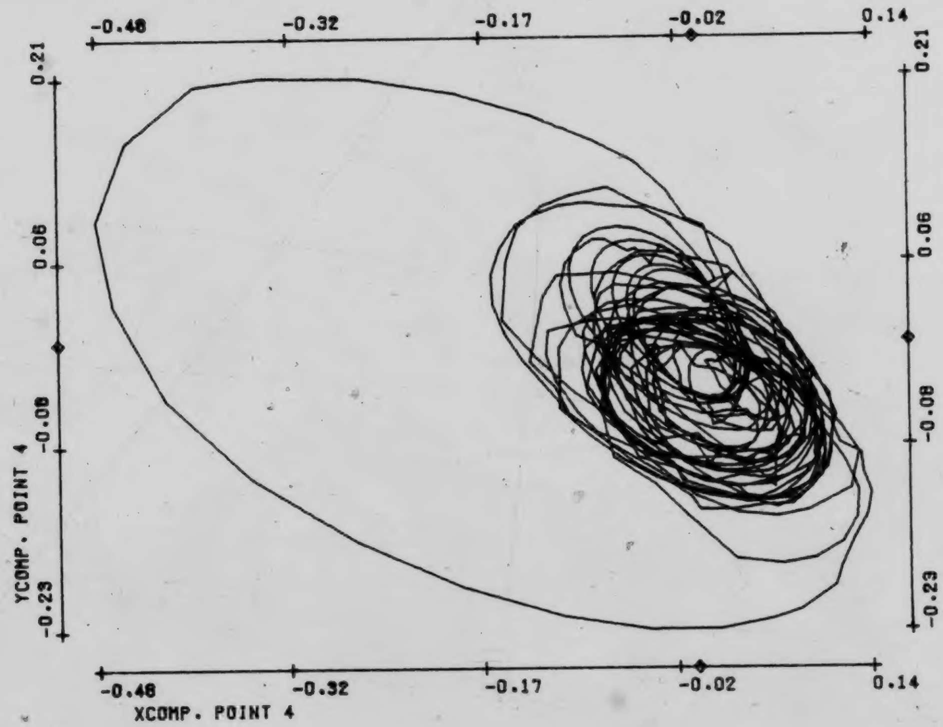
T13(1)


-0.40
-0.08
0.24
0.56
0.88
0.27
0.03
-0.21
-0.45
-0.69
YCOMP. POINT 1
XCOMP. POINT 1

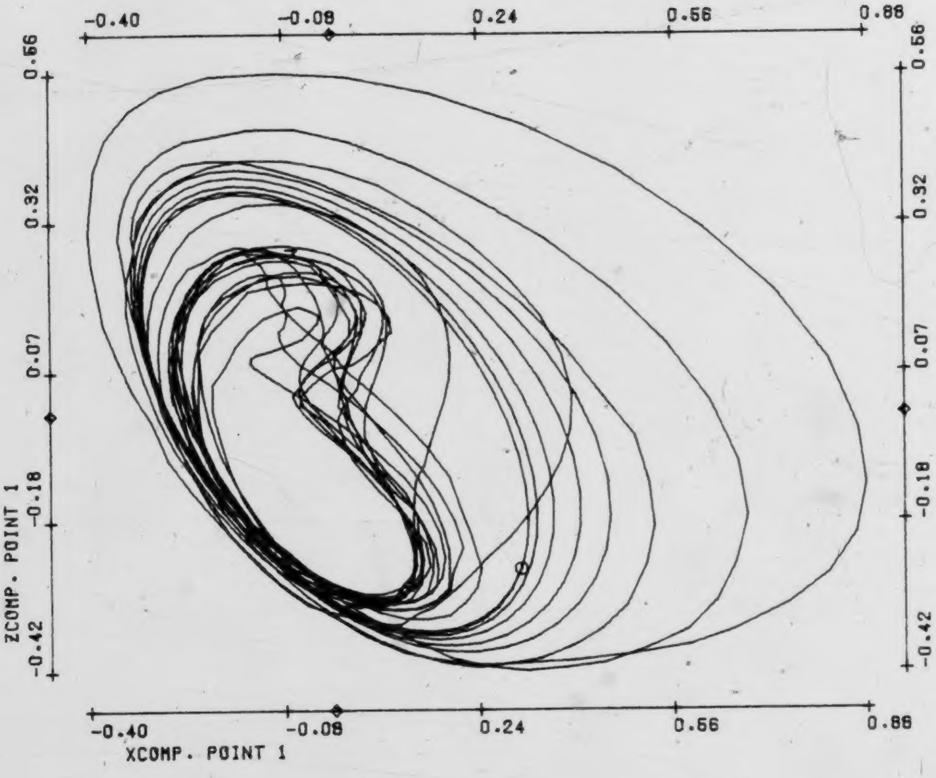

-0.10
0.03
0.15
0.28
0.40
0.11
-0.01
-0.13
YCOMP. POINT 2
XCOMP. POINT 2

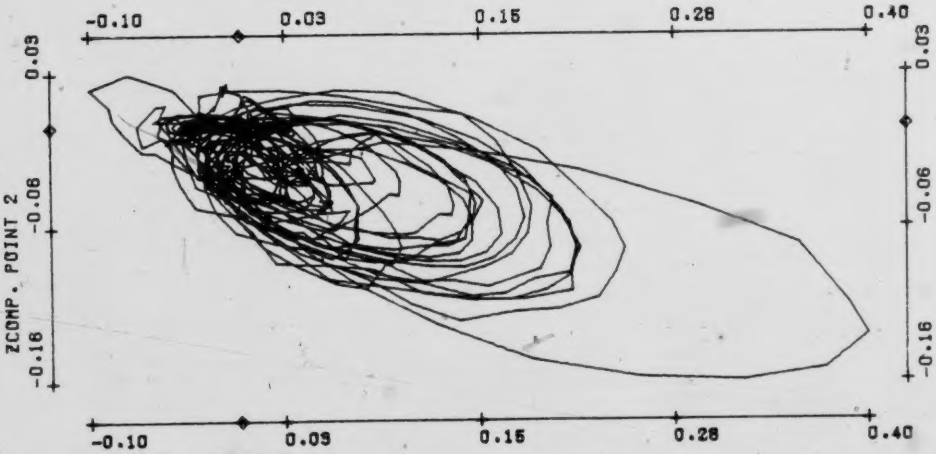
T13(2)

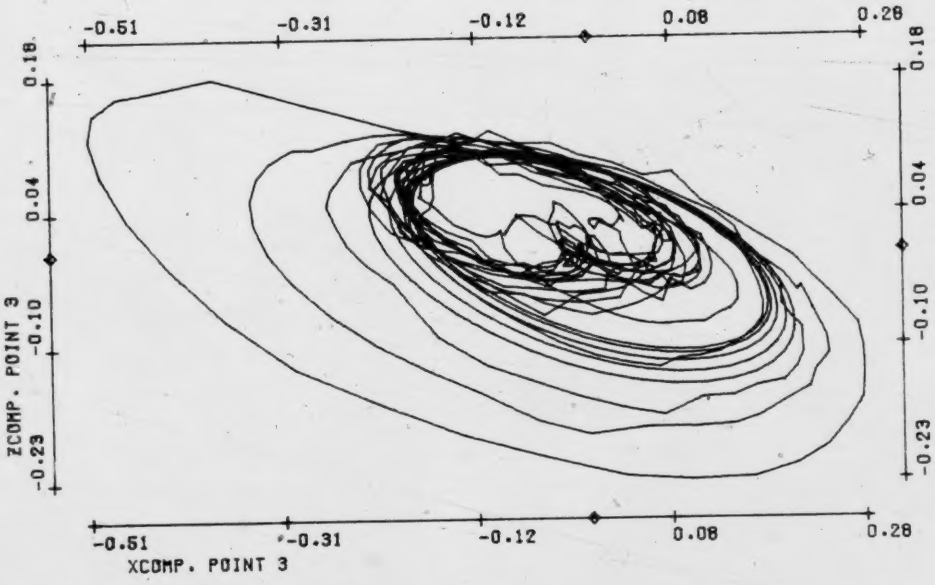

-0.51
-0.31
-0.12
0.08
0.28
0.36
0.20
0.04
-0.13
YCOMP. POINT 3
XCOMP. POINT 3

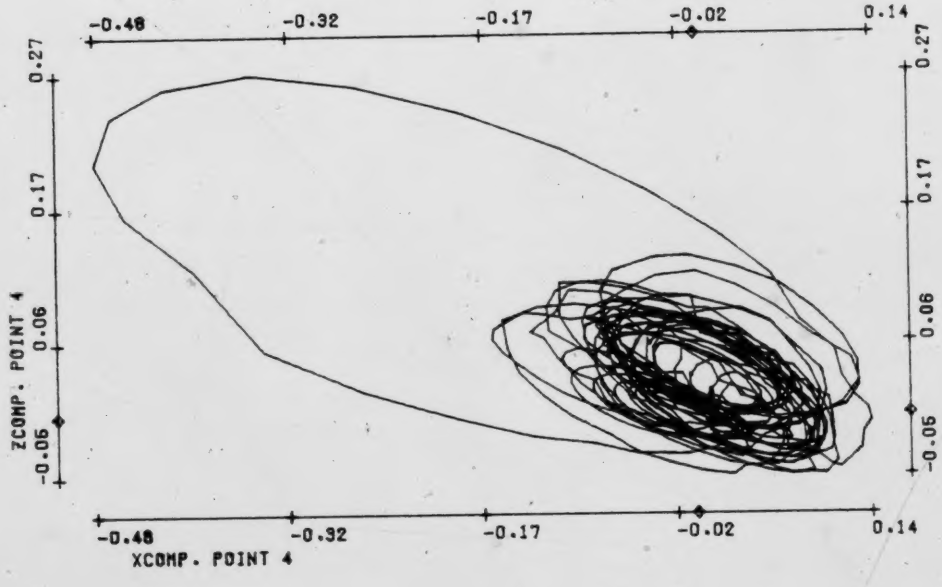

-0.48
-0.32
-0.17
-0.02
0.14
0.21
0.06
-0.08
-0.23
YCOMP. POINT 4
XCOMP. POINT 4

T13(3)

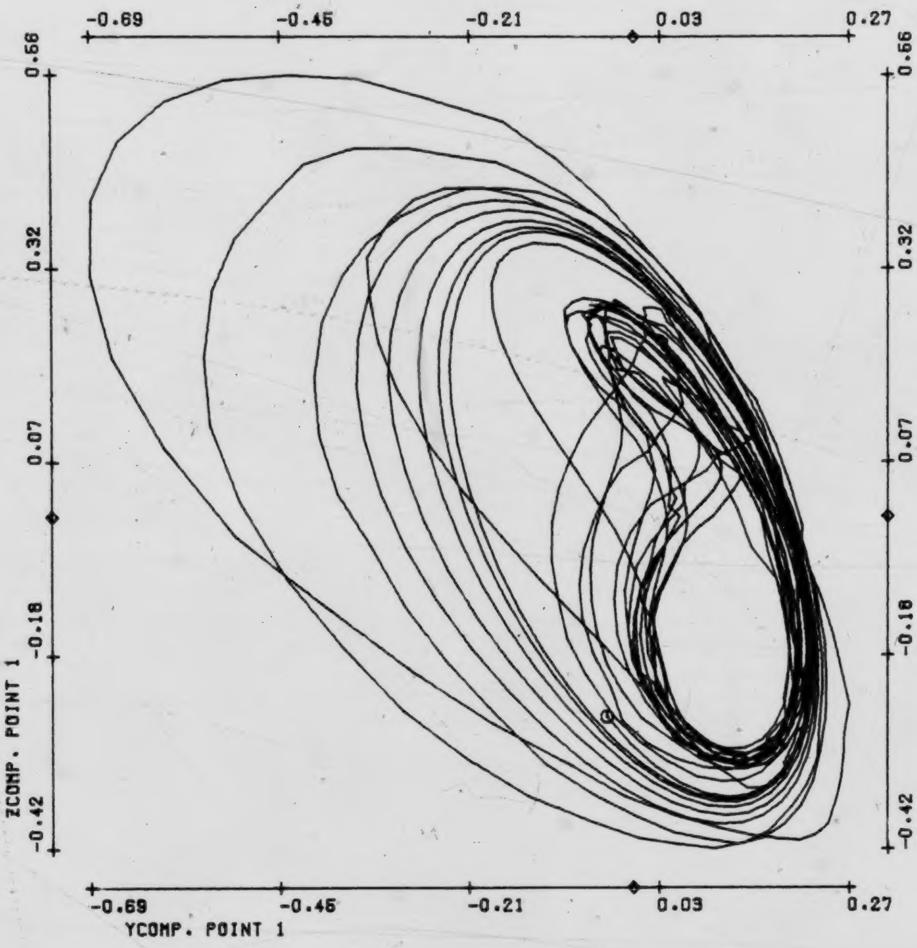

-0.40
-0.08
0.24
0.56
0.88
0.56
0.32
0.07
-0.18
-0.42
ZCOMP. POINT 1
XCOMP. POINT 1

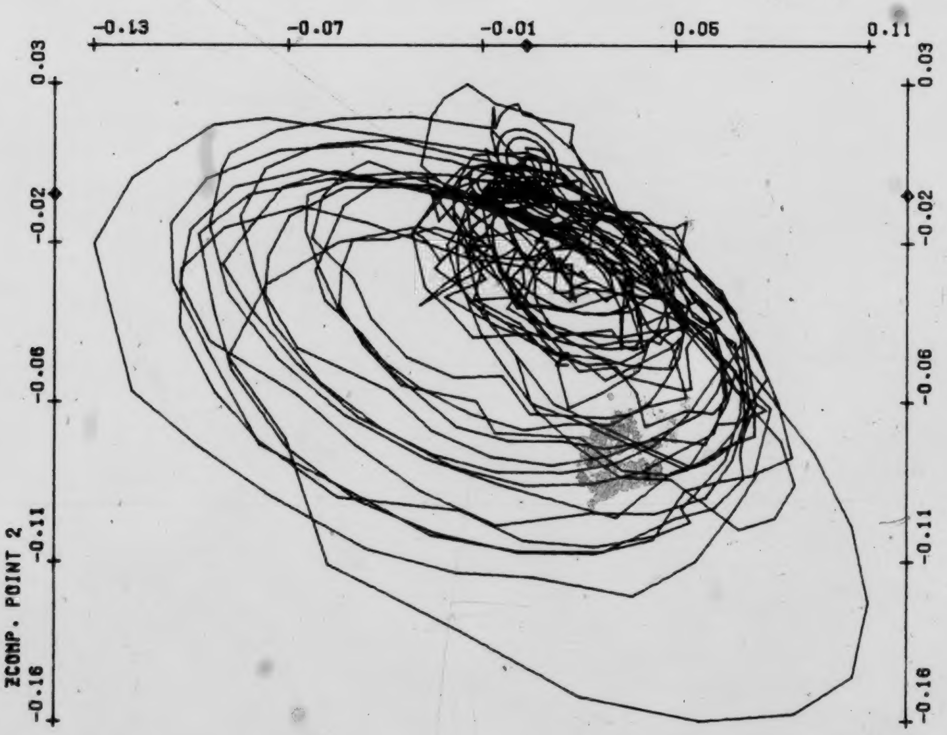

-0.10
0.03
0.15
0.28
0.40
0.03
-0.06
-0.16
ZCOMP. POINT 2
XCOMP. POINT 2


-0.51
-0.31
-0.12
0.08
0.28
0.18
0.04
-0.10
-0.23
ZCOMP. POINT 3
XCOMP. POINT 3

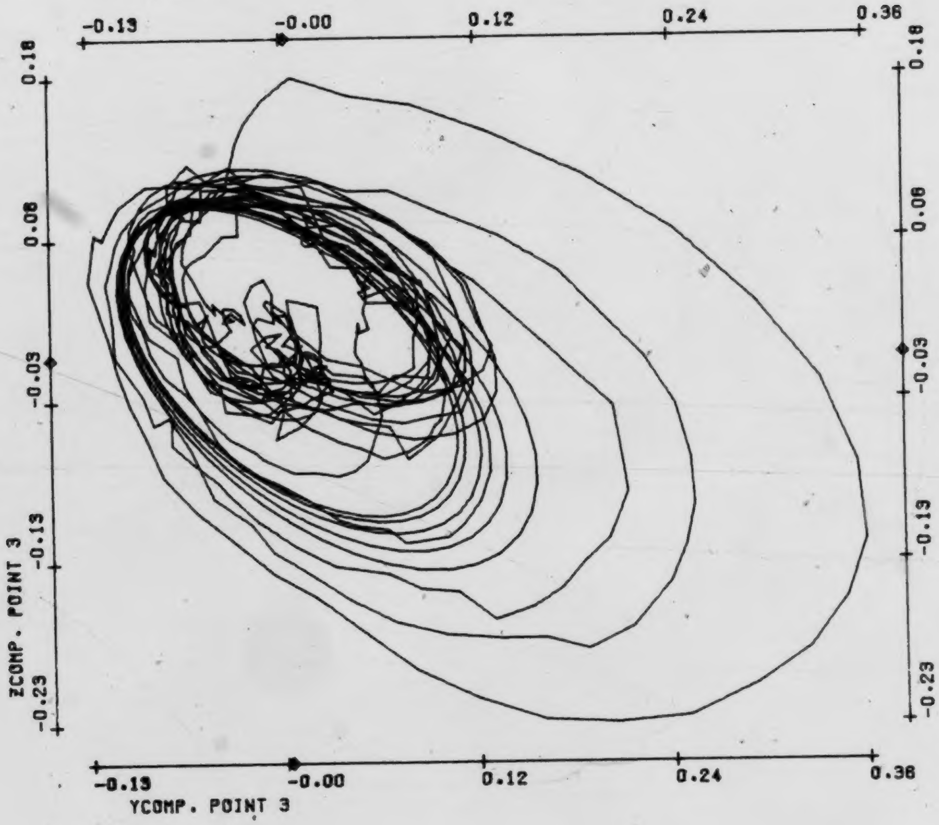

-0.48
-0.32
-0.17
-0.02
0.14
0.27
0.17
0.06
-0.05
ZCOMP. POINT 4
XCOMP. POINT 4

T13(5)


-0.69
-0.45
-0.21
0.03
0.27
0.56
0.32
0.07
-0.18
-0.42
ZCOMP. POINT 1
YCOMP. POINT 1


-0.13
-0.07
-0.01
0.05
0.11
0.03
-0.02
-0.06
-0.11
-0.16
ZCOMP. POINT 2
YCOMP. POINT 2

T13(6)


-0.13
-0.00
0.12
0.24
0.36
0.18
0.08
-0.03
-0.13
-0.23
ZCOMP. POINT 3
YCOMP. POINT 3

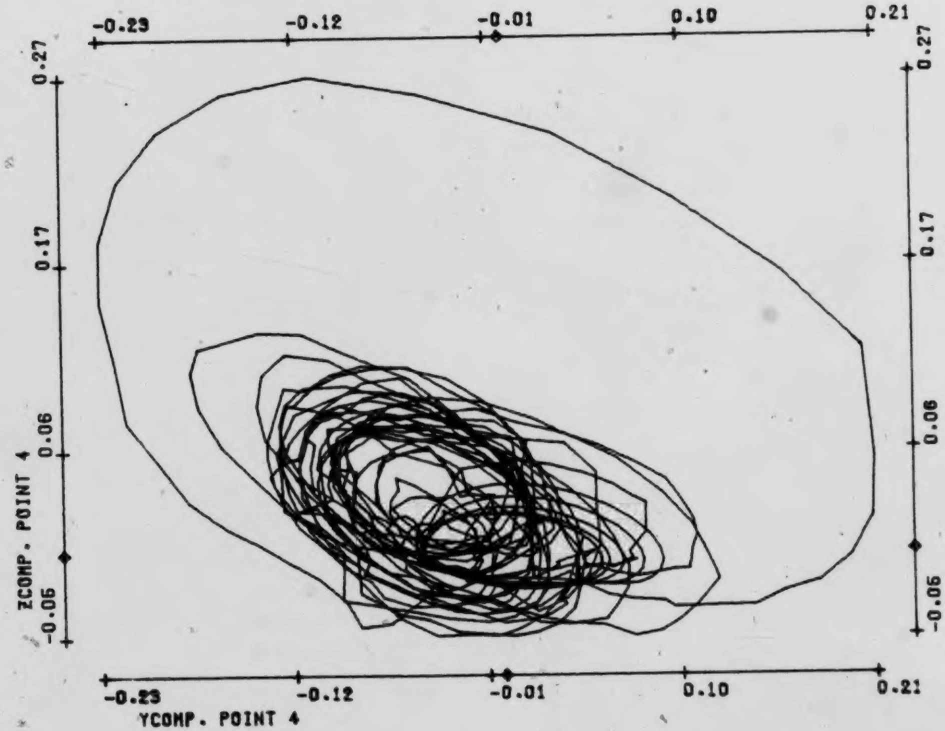
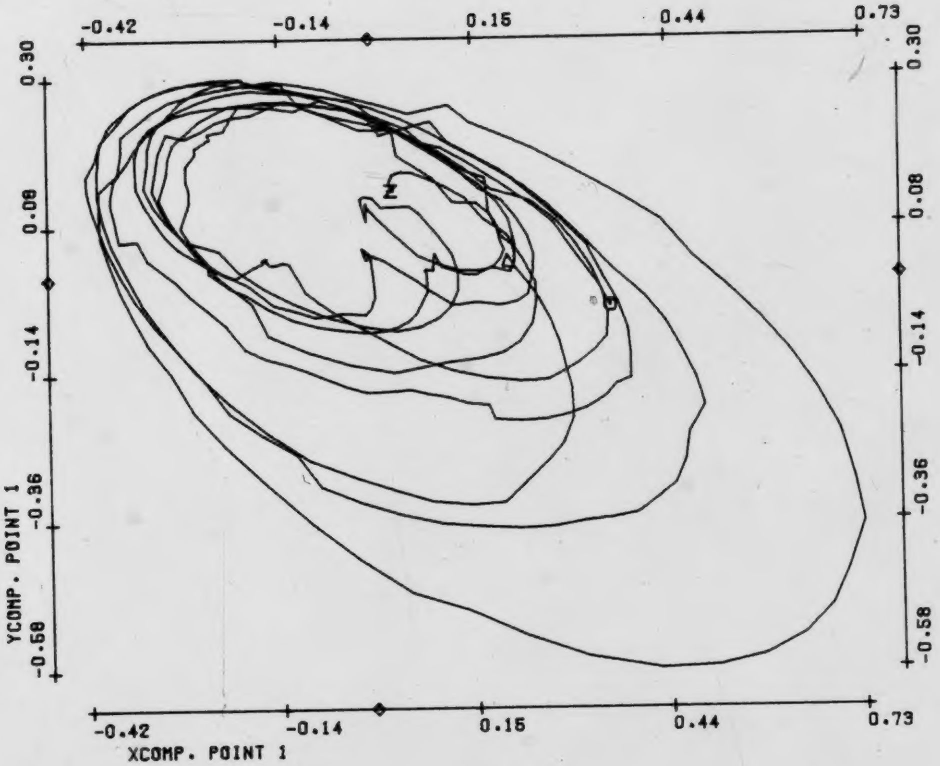
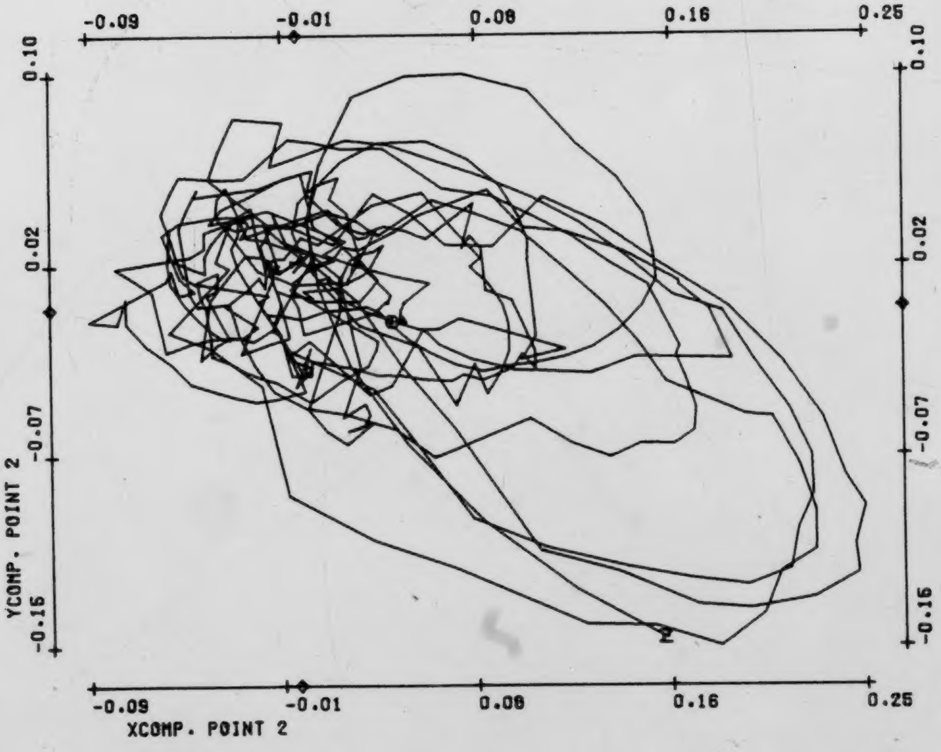

-0.23
-0.12
-0.01
0.10
0.21
0.27
0.17
0.06
-0.05
-0.06
ZCOMP. POINT 4
YCOMP. POINT 4

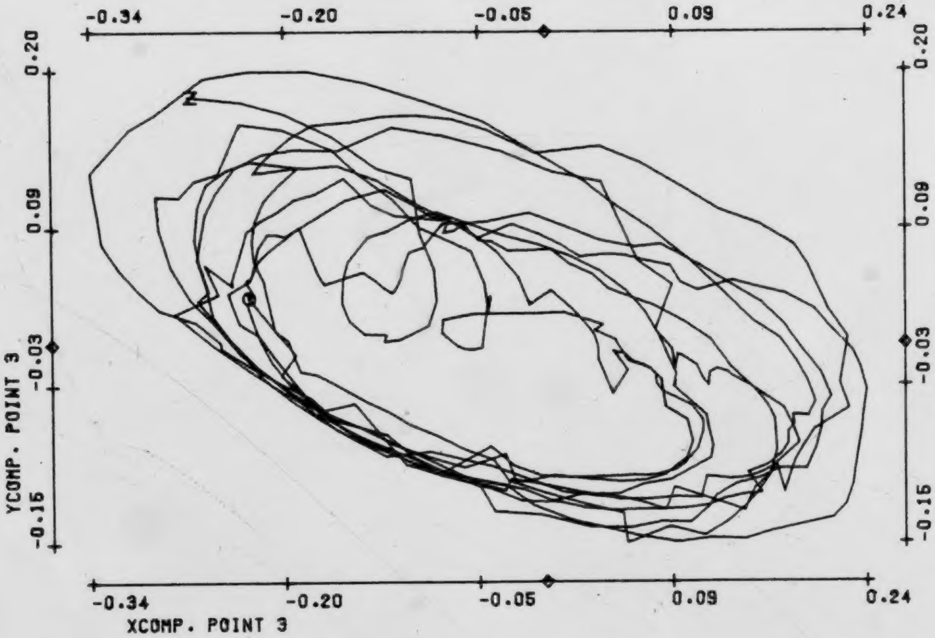
FIGURE T14

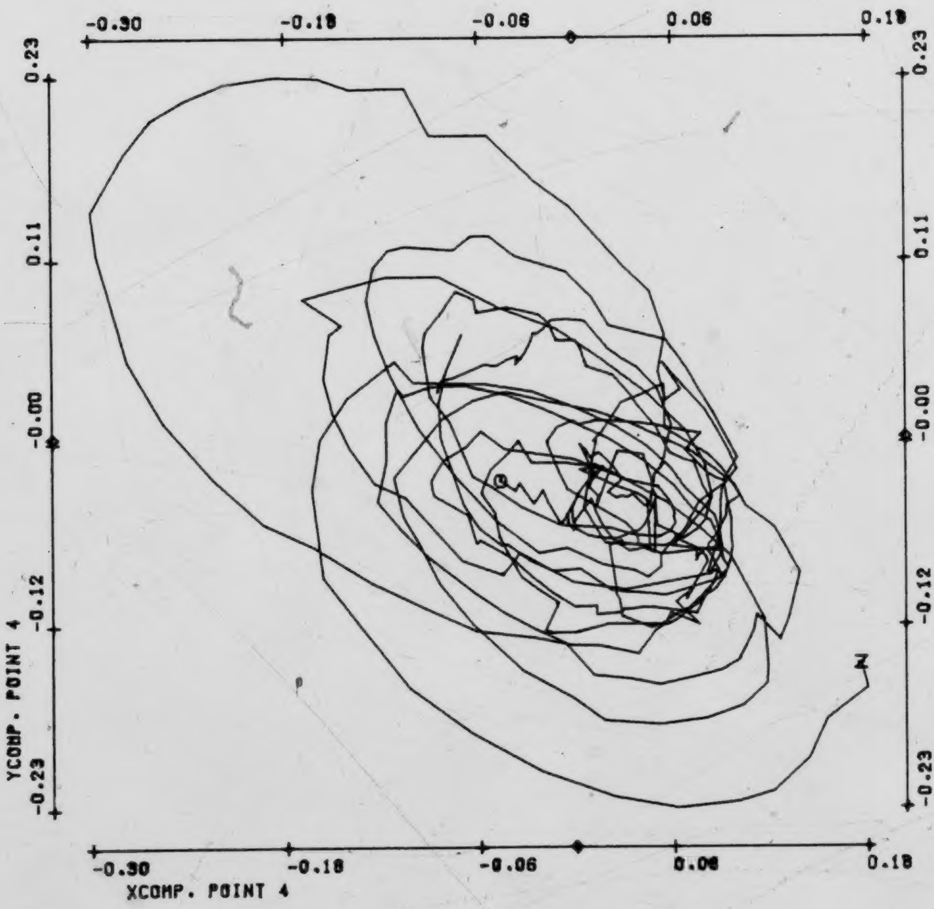
Bifurcation parameter η	.3
No. of timesteps	500
Stepsize	.03
Init. condition for $\vec{V}_2$	0(all)
Average size of spatial random perturbations	.02
Average no. of timesteps between random perturbations	2
Period of $\vec{V}_1$	1.68
Mean velocity:	
Point 1	2.20
2	2.03
3	2.95
4	1.85

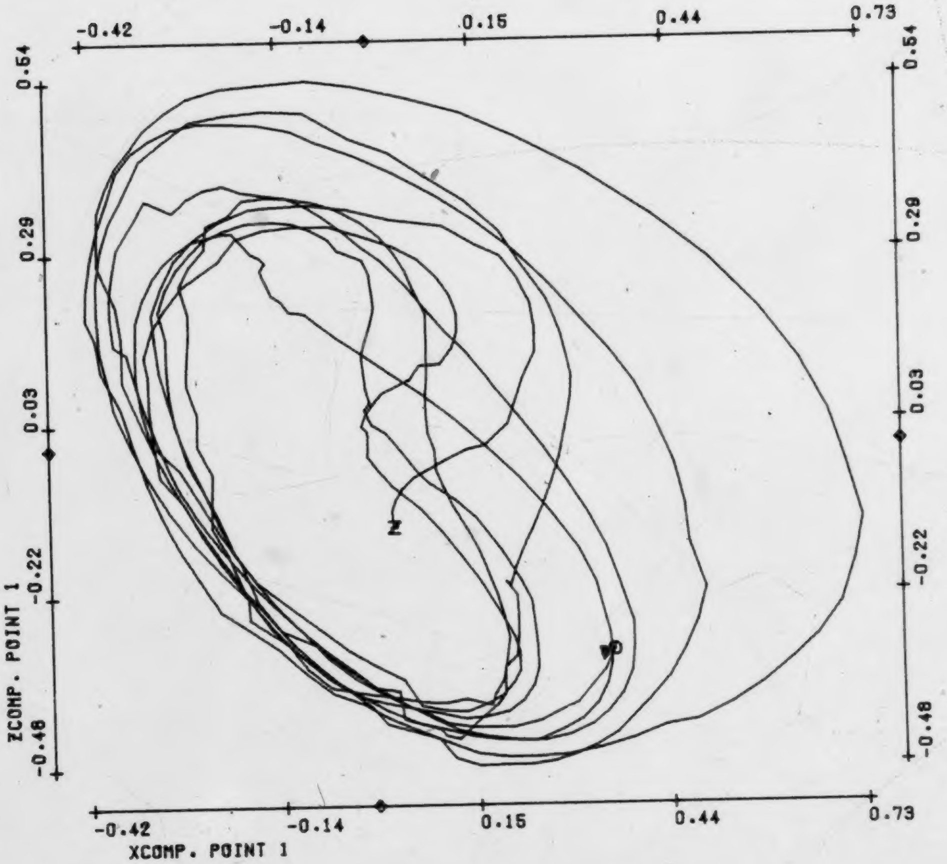
T14(1)

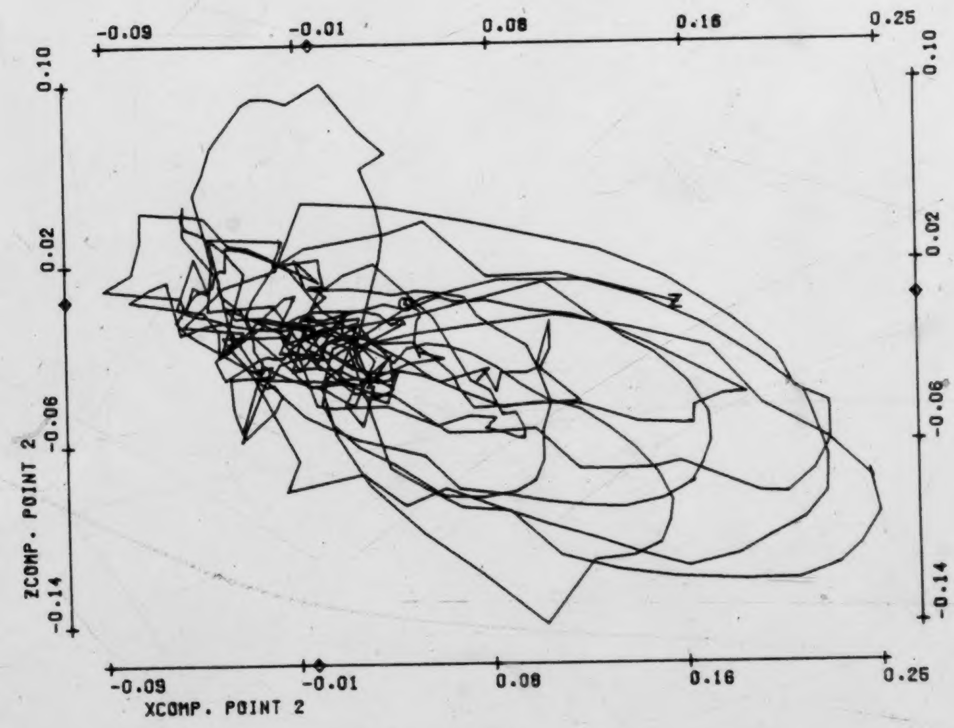

-0.42
-0.14
0.15
0.44
0.73
0.30
0.08
-0.14
-0.36
-0.58
YCOMP. POINT 1
XCOMP. POINT 1
Z
O

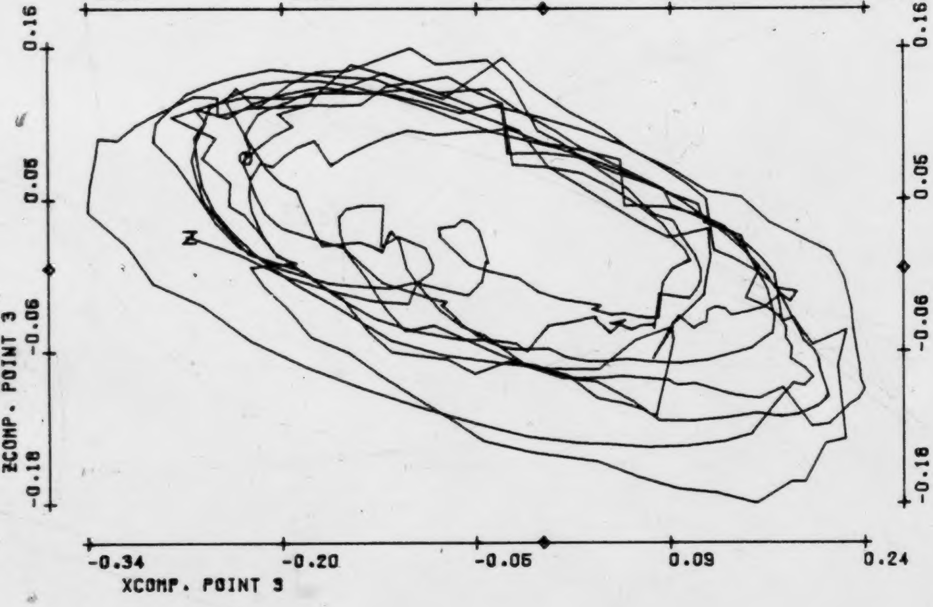

-0.09
-0.01
0.08
0.16
0.25
0.10
0.02
-0.07
-0.16
YCOMP. POINT 2
XCOMP. POINT 2
Z
O

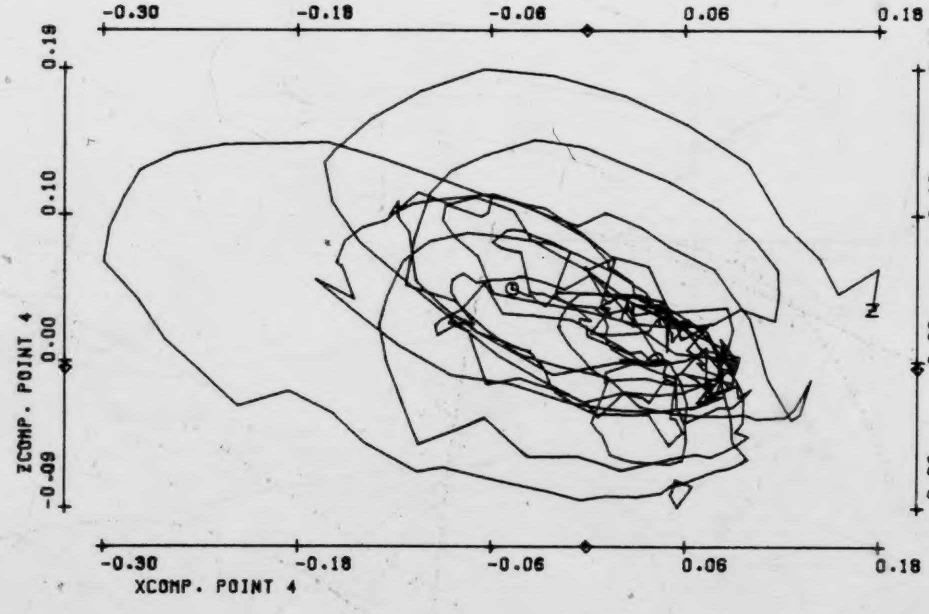

-0.34
-0.20
-0.05
0.09
0.24
0.20
0.09
-0.03
-0.15
Z
O
YCOMP. POINT 3
XCOMP. POINT 3


-0.30
-0.18
-0.06
0.06
0.18
0.23
0.11
-0.00
-0.12
-0.23
O
Z
YCOMP. POINT 4
XCOMP. POINT 4

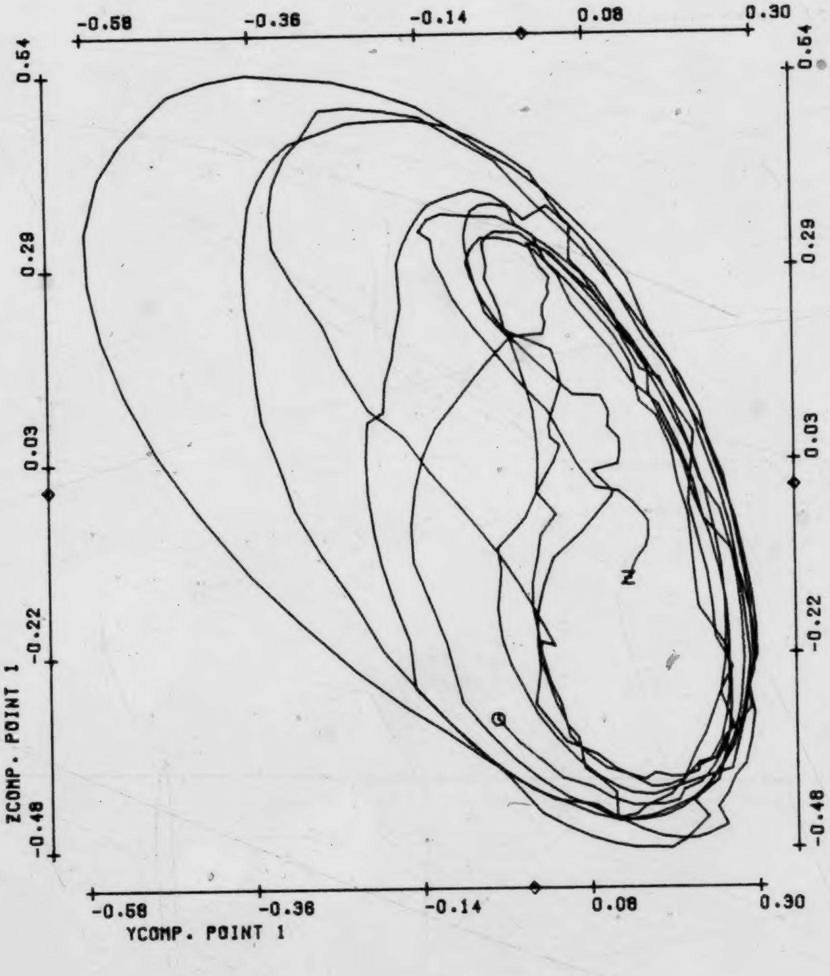

-0.42
-0.14
0.15
0.44
0.73
0.54
0.29
0.03
-0.22
-0.48
Z
A
ZCOMP. POINT 1
XCOMP. POINT 1

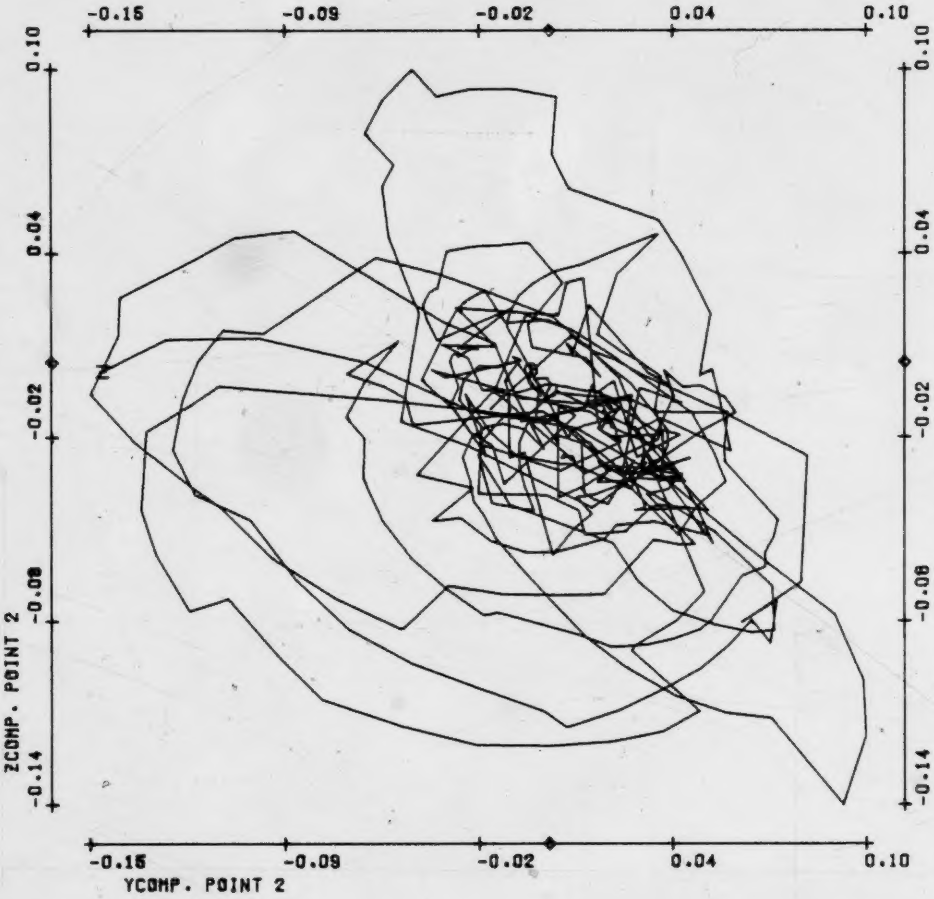

-0.09
-0.01
0.08
0.16
0.25
0.10
0.02
-0.06
-0.14
Z
ZCOMP. POINT 2
XCOMP. POINT 2


-0.34
-0.20
-0.05
0.09
0.24
0.16
0.05
-0.06
-0.18
ZCOMP. POINT 3
XCOMP. POINT 3
Z
O

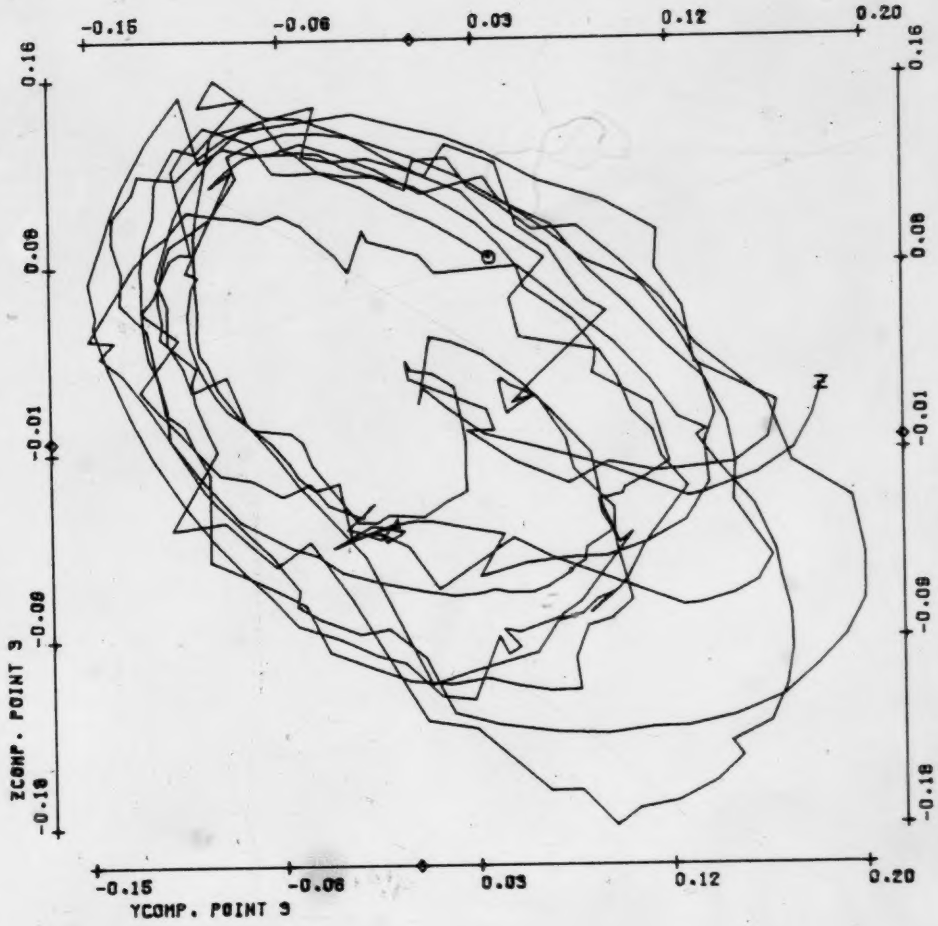

-0.30
-0.18
-0.06
0.06
0.18
0.19
0.10
0.00
-0.09
ZCOMP. POINT 4
XCOMP. POINT 4
Z
O

T14(5)


-0.58
-0.36
-0.14
0.08
0.30
0.54
0.29
0.03
-0.22
-0.48
Z
O
ZCOMP. POINT 1
YCOMP. POINT 1


-0.15
-0.09
-0.02
0.04
0.10
0.10
0.04
-0.02
-0.08
-0.14
ZCOMP. POINT 2
YCOMP. POINT 2

T14(7)


-0.15
-0.06
0.03
0.12
0.20
0.16
0.08
-0.01
-0.09
-0.18
ZCOMP. POINT 3
YCOMP. POINT 3
Z
G

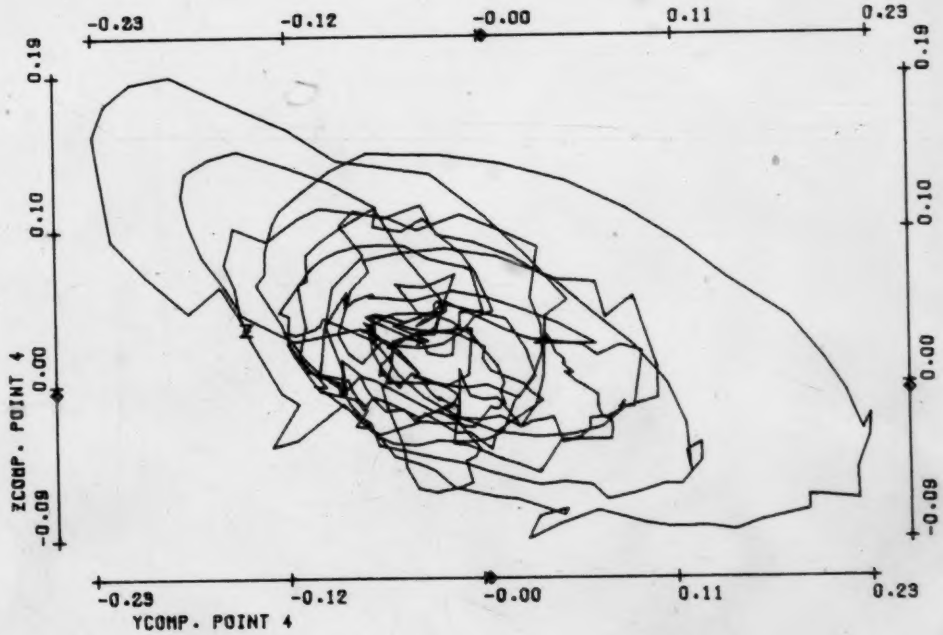
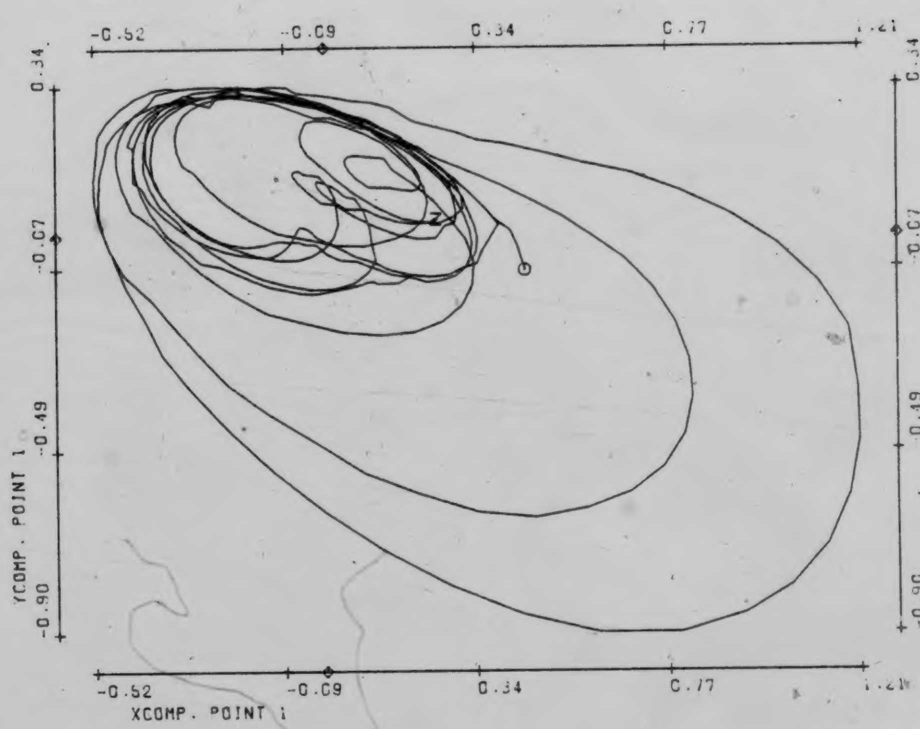
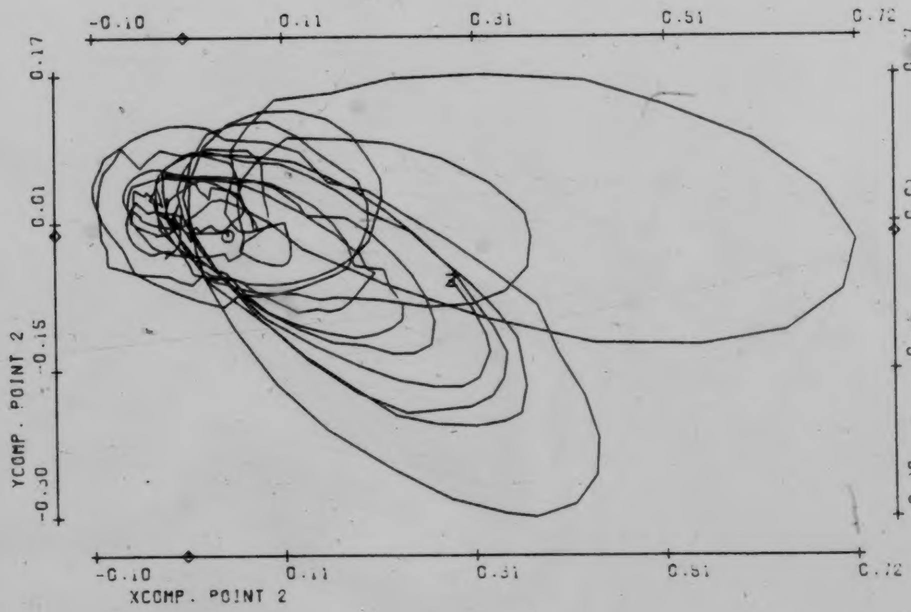

-0.23
-0.12
-0.00
0.11
0.23
0.19
0.10
0.00
-0.09
ZCOMP. POINT 4
YCOMP. POINT 4
Z

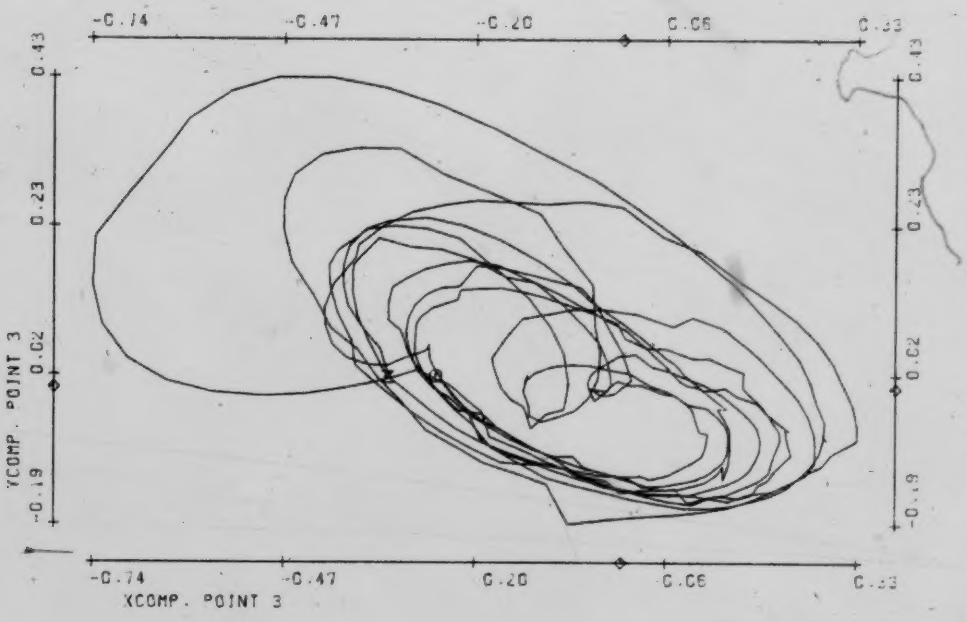
FIGURE T15

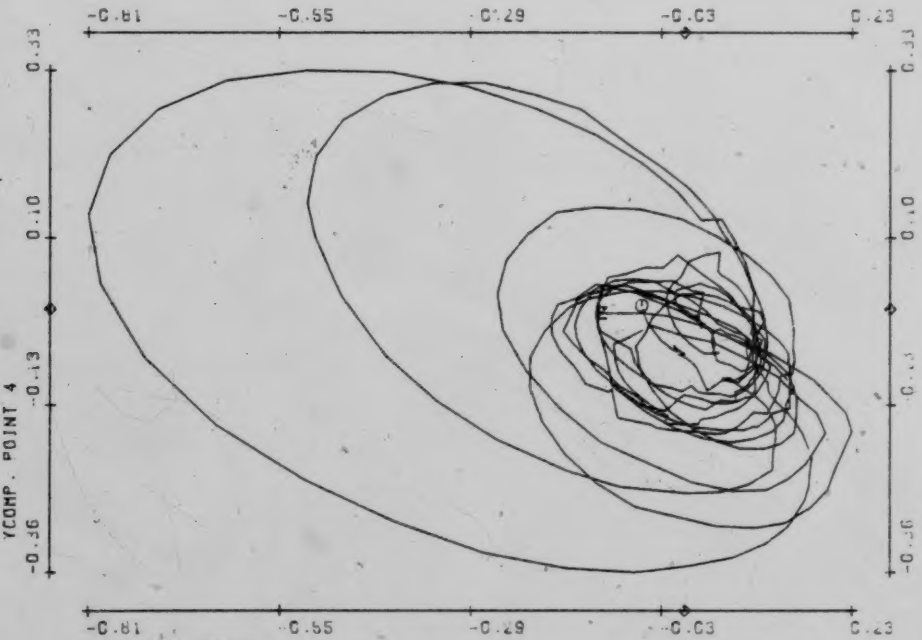
Bifurcation parameter η	.325
No. of timesteps	500
Stepsize	.03
Init. condition for $\vec{V}_2$	0 (all)
Average size of spatial random perturbations	.015
Average no. of timesteps between random perturbations	2.5
Period of $\vec{V}_1$	1.66
Mean velocity:	
Point 1	1.95
2	1.64
3	2.65
4	1.60


-0.52
-0.09
0.34
0.77
1.21
0.34
-0.07
-0.49
-0.90
YCOMP. POINT 1
XCOMP. POINT 1
Z

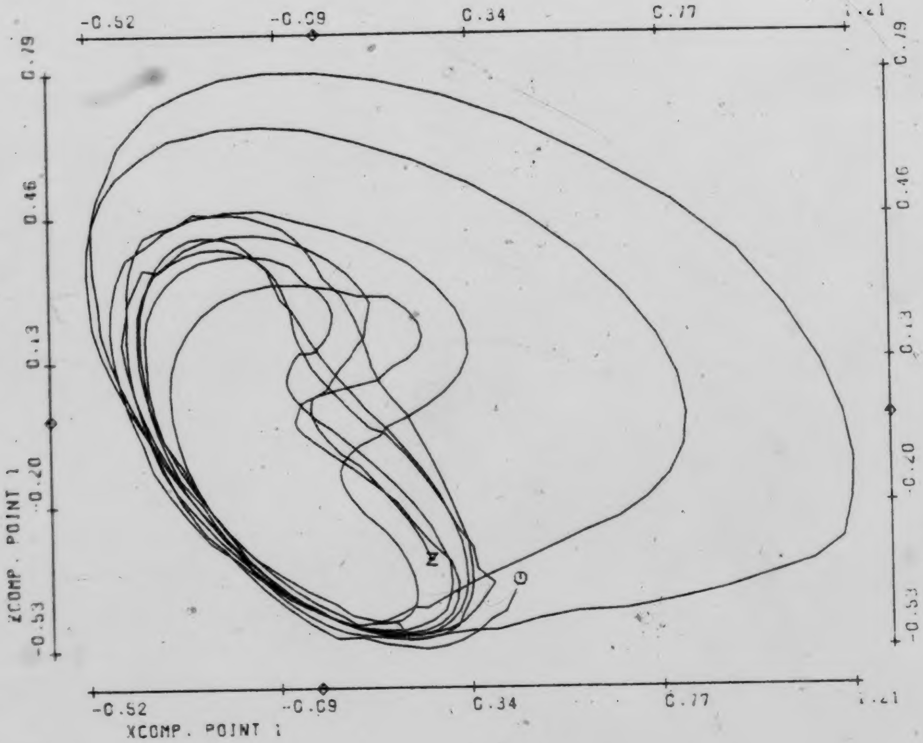

-0.10
0.11
0.31
0.51
0.72
0.17
0.01
-0.15
-0.30
YCOMP. POINT 2
XCOMP. POINT 2
Z

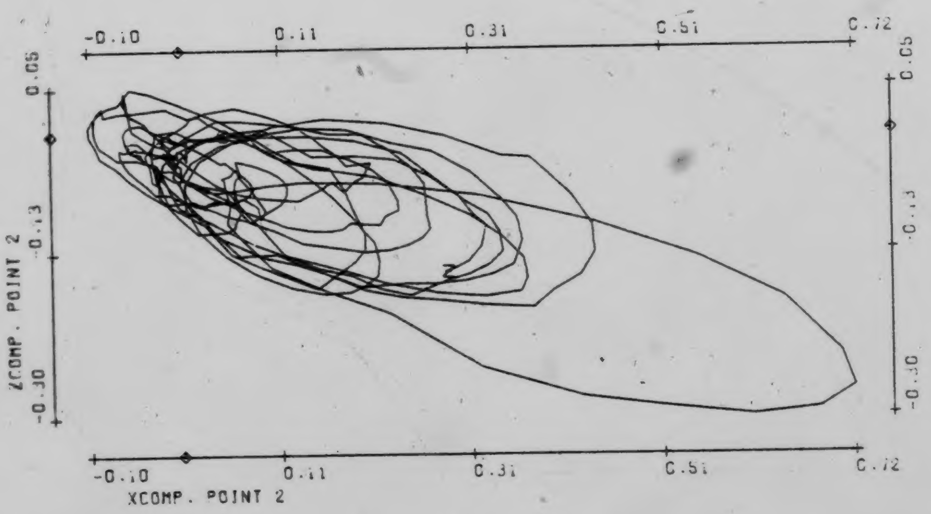
T15(2)


-0.74
-0.47
-0.20
0.06
0.33
0.43
0.23
0.02
-0.19
YCOMP. POINT 3
XCOMP. POINT 3

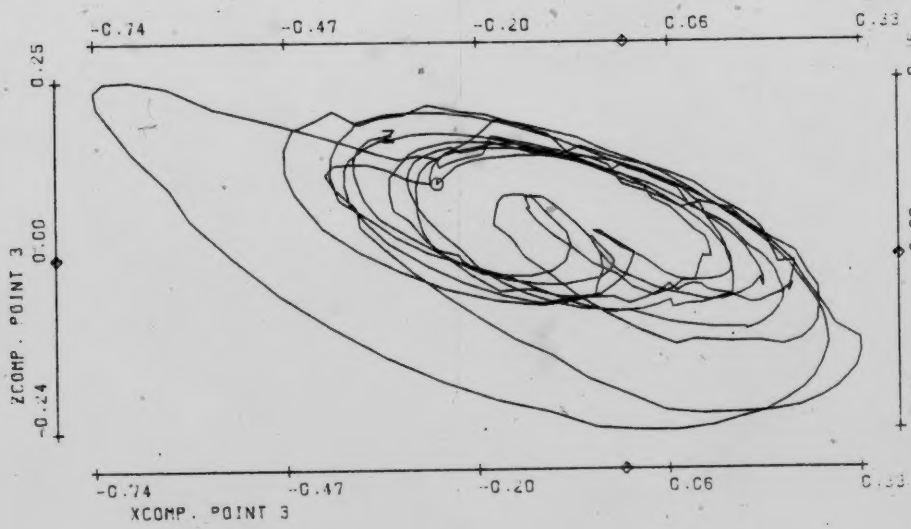

-0.81
-0.55
-0.29
-0.03
0.23
0.33
0.10
-0.13
-0.36
YCOMP. POINT 4
XCOMP. POINT 4

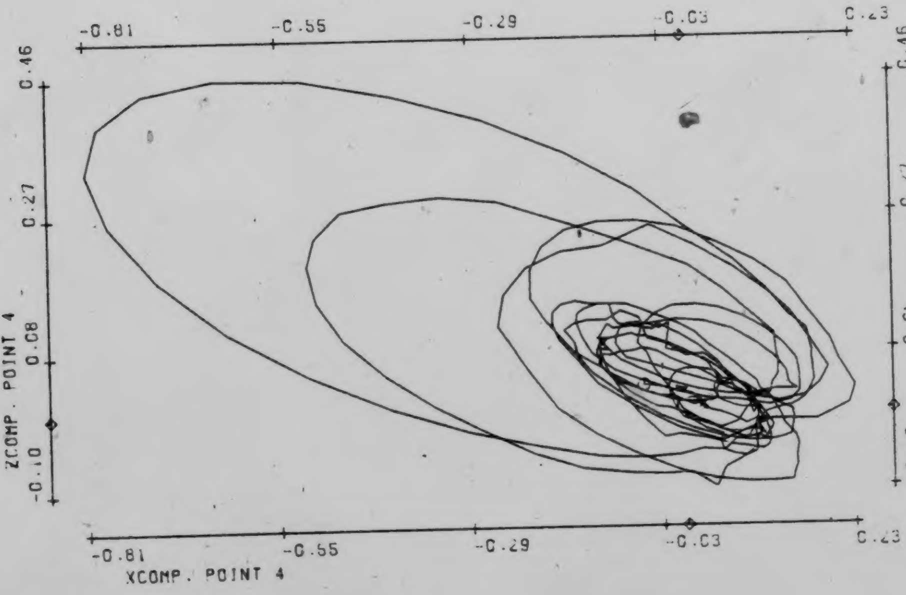
T15(3)


-0.52
-0.09
0.34
0.77
1.21
0.79
0.46
0.13
-0.20
-0.53
ZCOMP. POINT 1
XCOMP. POINT 1

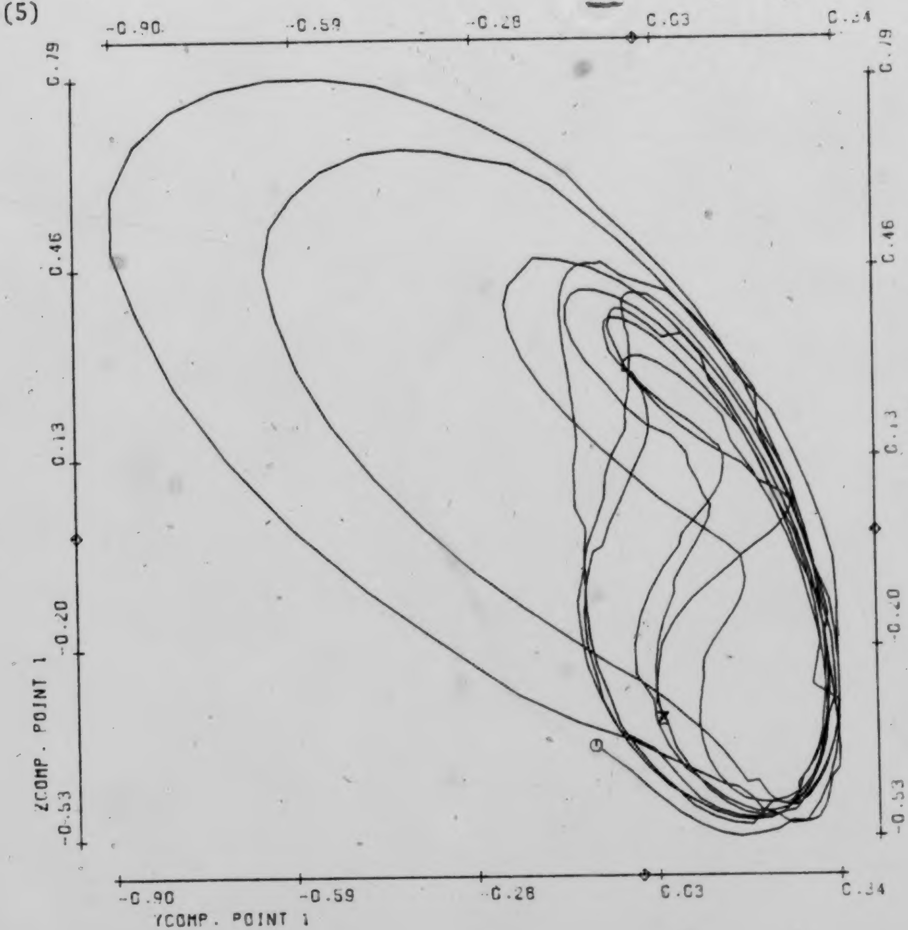

-0.10
0.11
0.31
0.51
0.72
0.05
-0.13
-0.30
ZCOMP. POINT 2
XCOMP. POINT 2

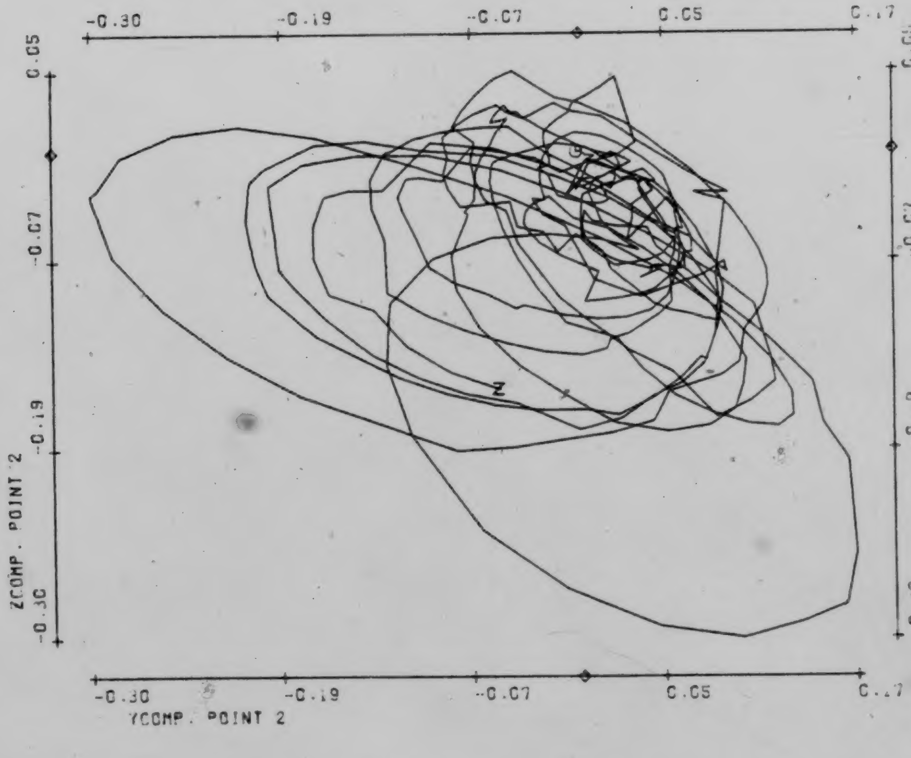
T15(4)


-0.74
-0.47
-0.20
0.06
0.33
0.25
0.00
-0.24
ZCOMP. POINT 3
XCOMP. POINT 3


-0.81
-0.55
-0.29
-0.03
0.23
0.46
0.27
0.08
-0.10
ZCOMP. POINT 4
XCOMP. POINT 4

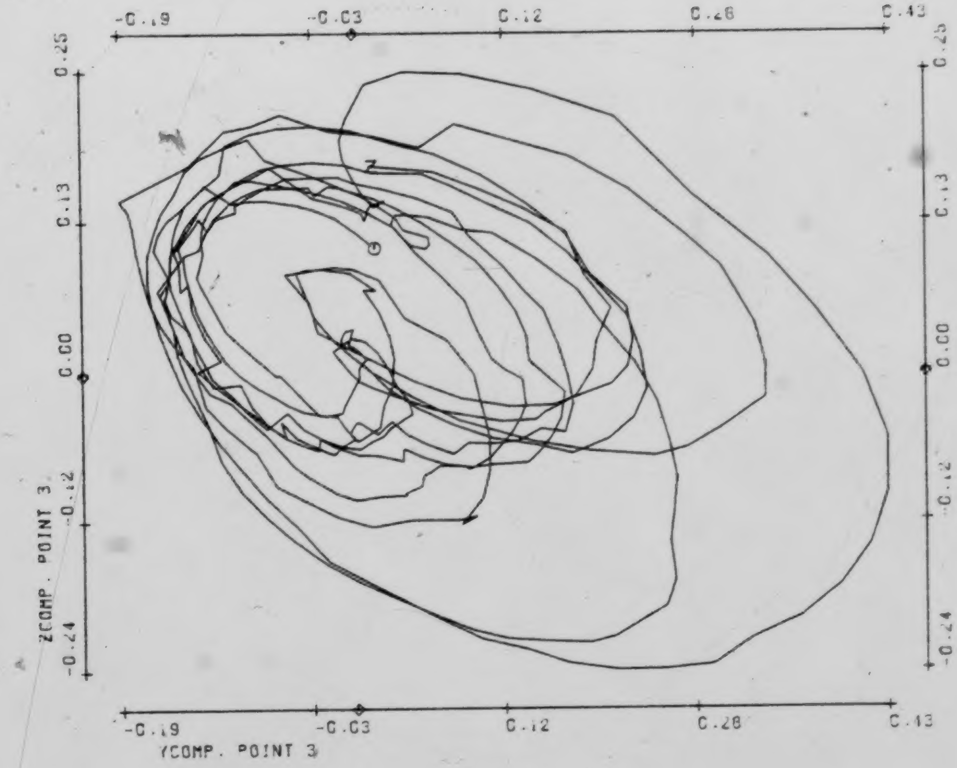
T15(5)


-0.90
-0.59
-0.28
0.03
0.34
0.79
0.46
0.13
-0.20
-0.53
ZCOMP. POINT 1
YCOMP. POINT 1


-0.30
-0.19
-0.07
0.05
0.17
0.05
-0.07
-0.19
-0.30
ZCOMP. POINT 2
YCOMP. POINT 2

T15(6)

7


-0.19
-0.03
0.12
0.28
0.43
0.25
0.13
0.00
-0.12
-0.24
ZCOMP. POINT 3
YCOMP. POINT 3

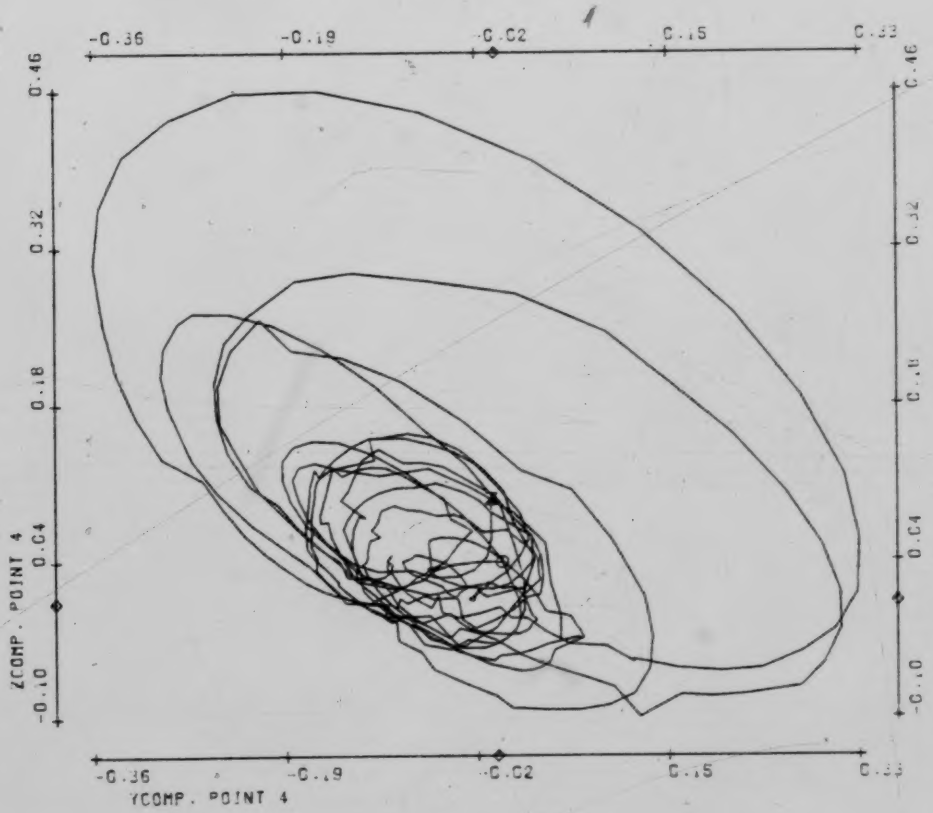
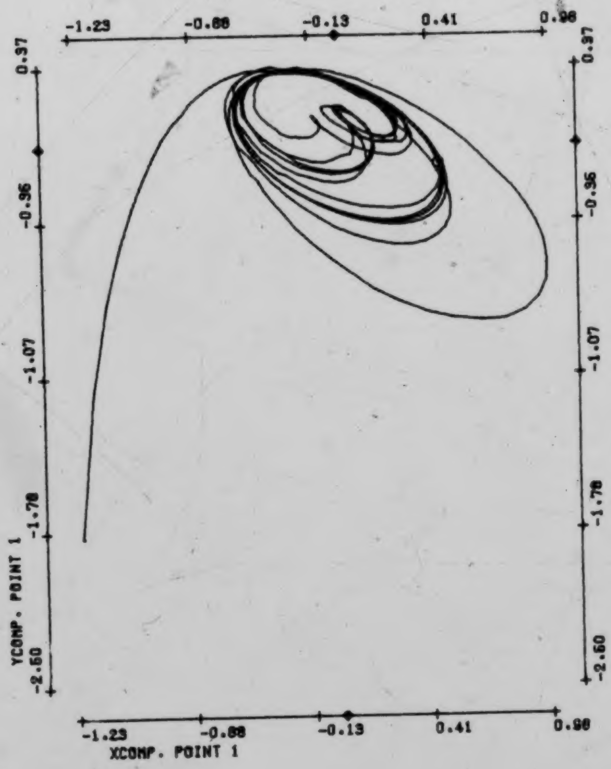

-0.36
-0.19
-0.02
0.15
0.33
0.46
0.32
0.18
0.04
-0.10
ZCOMP. POINT 4
YCOMP. POINT 4

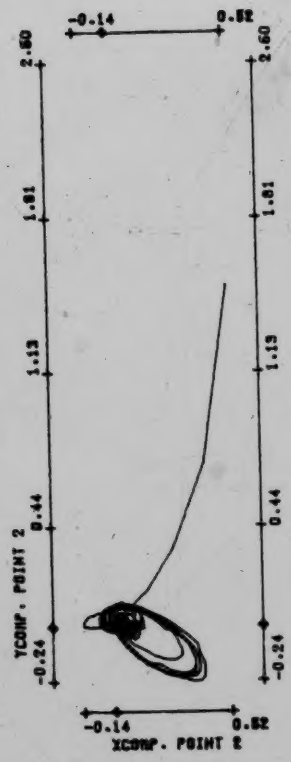
FIGURE T16

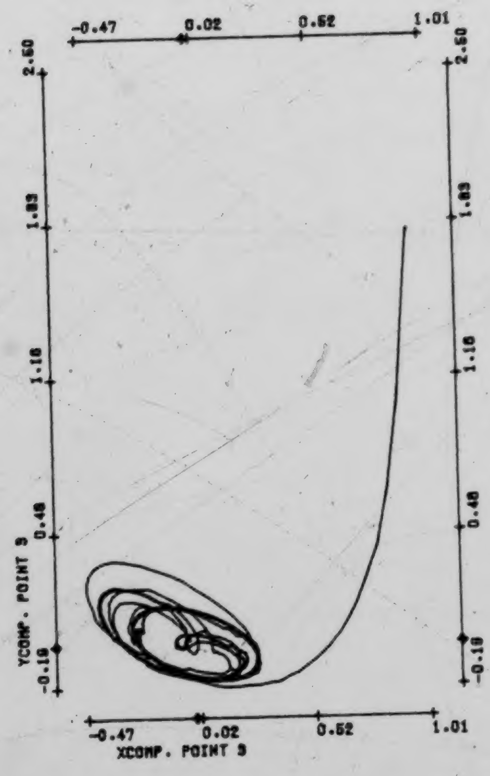
Bifurcation parameter η	.33
No. of timesteps	645
Stepsize	.03
Init. condition for $\vec{V}_2$	.001 (X-comp.Pt.1) rest = 0
Average size of spatial random perturbations	.01
Average no. of timesteps between random perturbations	2
Period of $\vec{V}_1$	1.65
Mean velocity:	
Point 1	1.75
2	1.58
3	2.43
4	1.59

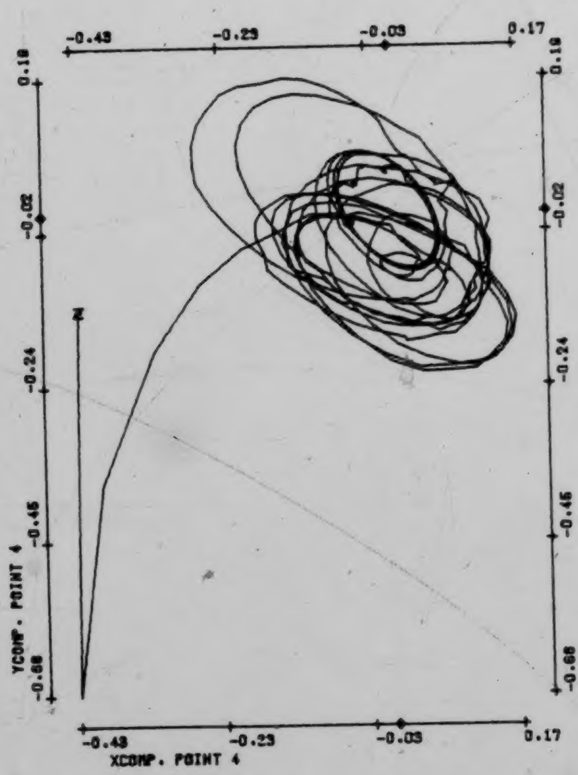
First 200 timesteps are unperturbed.

T16(1)

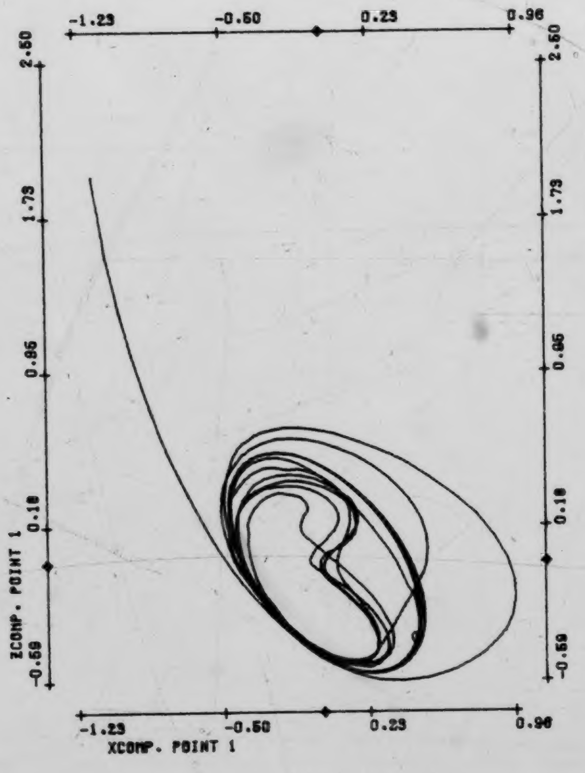

-1.23
-0.68
-0.13
0.41
0.96
0.37
-0.36
-1.07
-1.78
-2.50
YCOMP. POINT 1
XCOMP. POINT 1

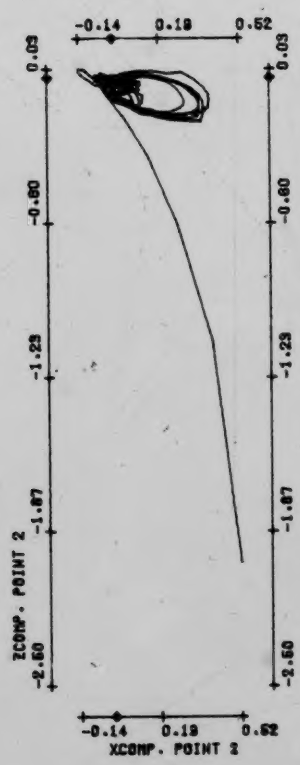

-0.14
0.52
2.50
1.81
1.13
0.44
-0.24
YCOMP. POINT 2
XCOMP. POINT 2


-0.47
0.02
0.52
1.01
2.50
1.83
1.16
0.49
-0.18
YCOMP. POINT 3
XCOMP. POINT 3

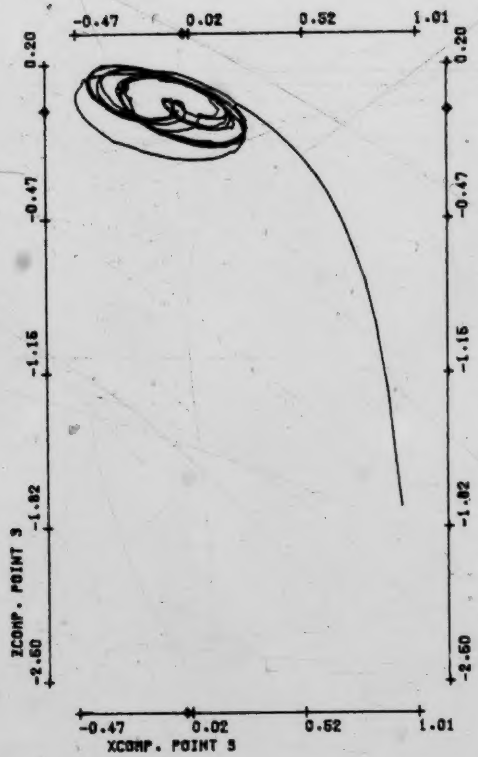

-0.43
-0.23
-0.03
0.17
0.19
-0.02
-0.24
-0.45
-0.66
N
YCOMP. POINT 4
XCOMP. POINT 4

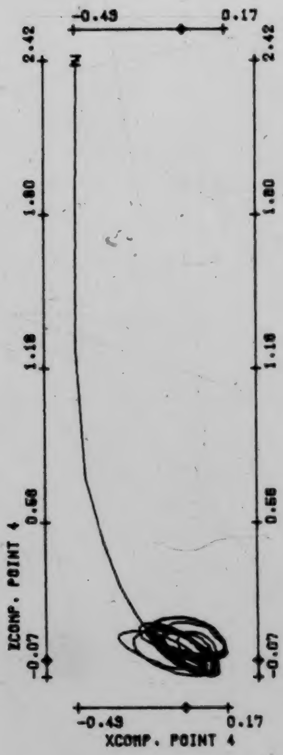
T16(3)


-1.23
-0.50
0.23
0.96
2.50
1.73
0.96
0.18
-0.59
ZCOMP. POINT 1
XCOMP. POINT 1

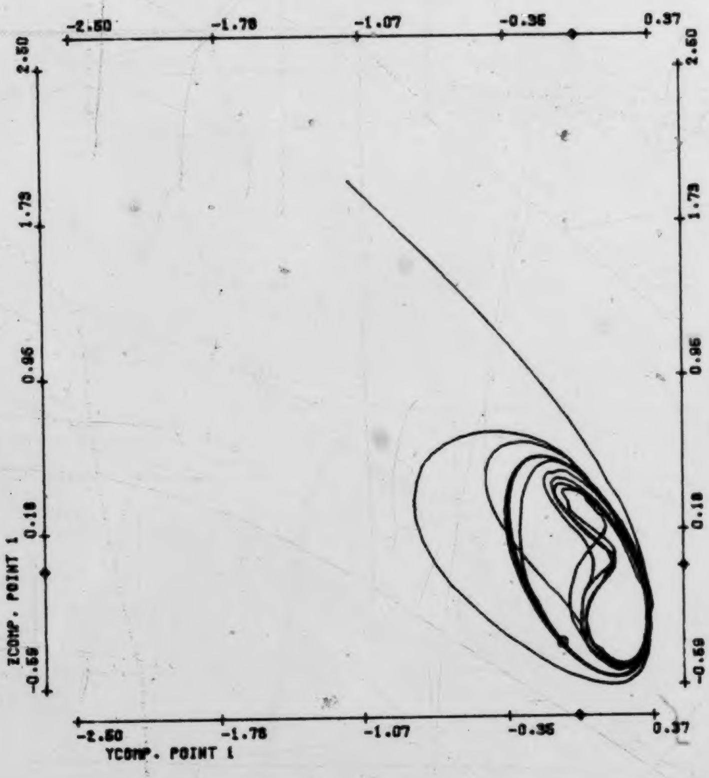

-0.14
0.19
0.52
0.03
-0.60
-1.23
-1.87
-2.50
ZCOMP. POINT 2
XCOMP. POINT 2

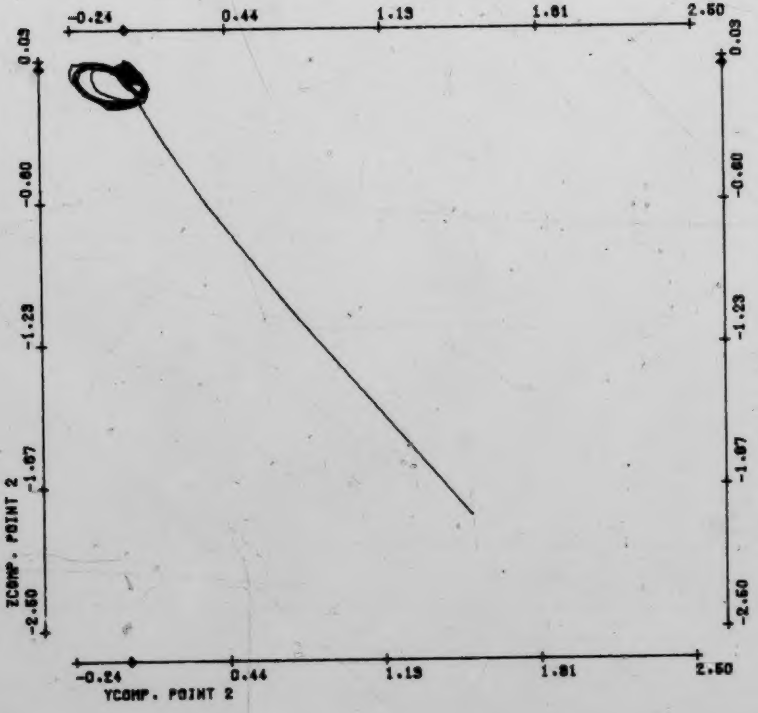
T16(4)

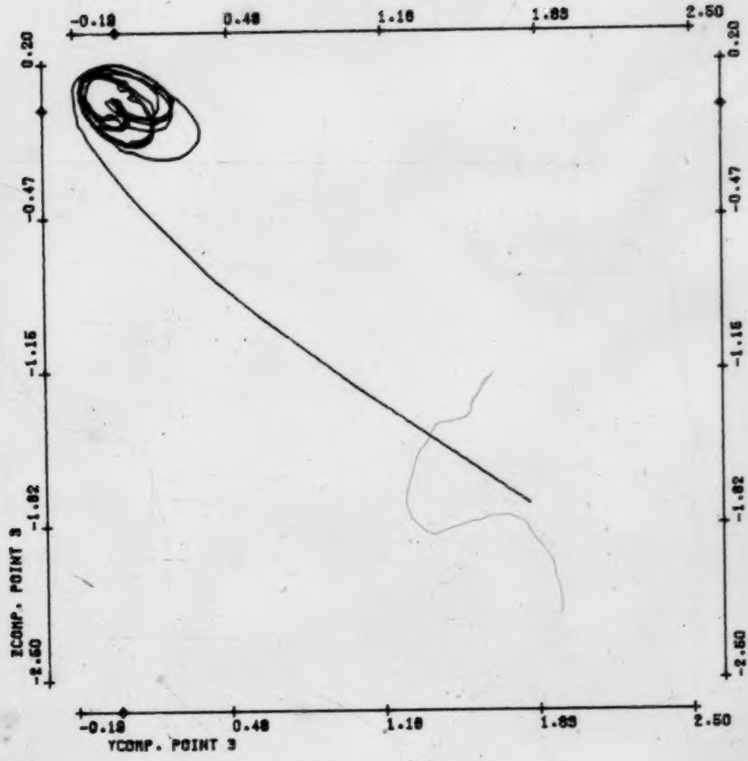

-0.47
0.02
0.52
1.01
0.20
-0.47
-1.15
-1.82
-2.50
ZCOMP. POINT 3
XCOMP. POINT 3


-0.43
0.17
2.42
1.80
1.18
0.56
-0.07
ZCOMP. POINT 4
XCOMP. POINT 4

T16(5)


-2.50
-1.78
-1.07
-0.35
0.37
2.50
1.73
0.96
0.18
-0.59
ZCOMP. POINT 1
YCOMP. POINT 1


-0.24
0.44
1.13
1.81
2.50
0.03
-0.60
-1.23
-1.87
-2.50
ZCOMP. POINT 2
YCOMP. POINT 2


-0.19
0.48
1.16
1.83
2.50
0.20
-0.47
-1.15
-1.82
-2.50
ZCOMP. POINT 3
YCOMP. POINT 3

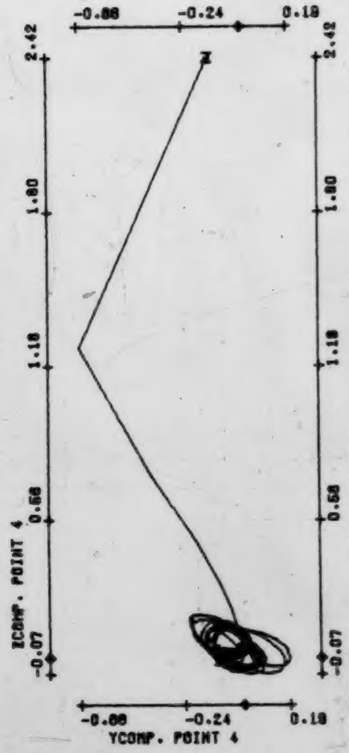
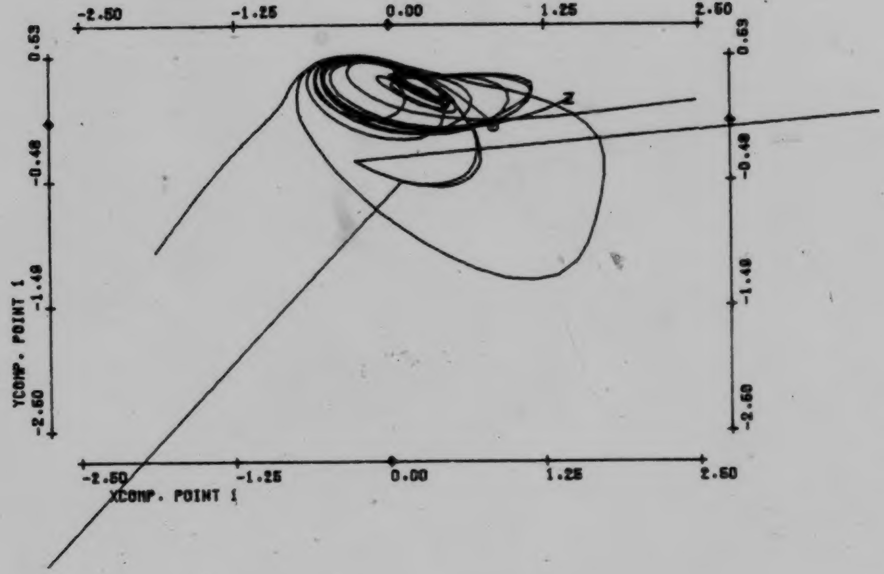
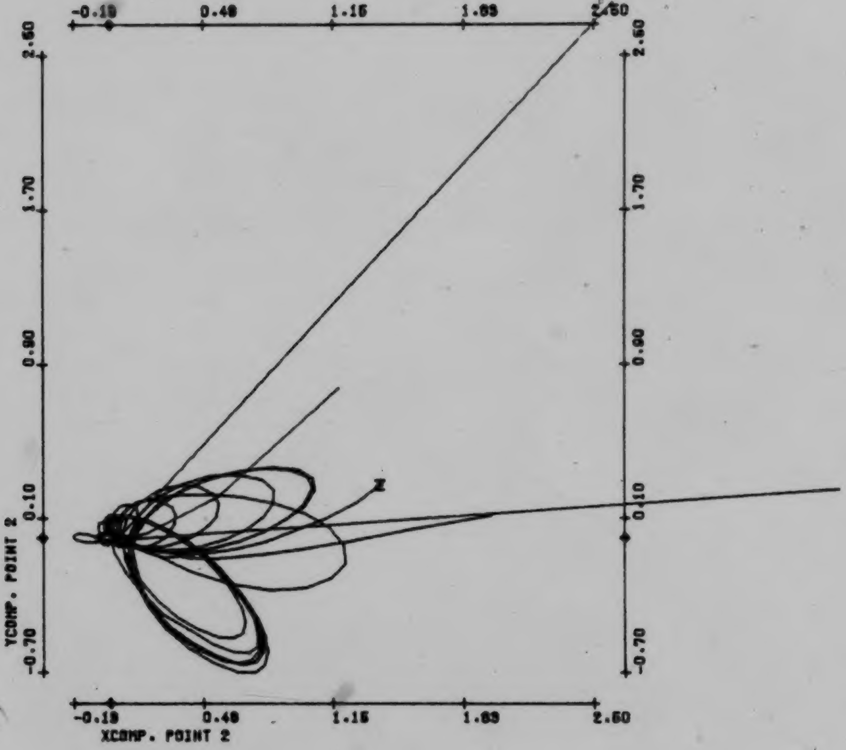

-0.66
-0.24
0.18
2.42
1.80
1.18
0.56
-0.07
ZCOMP. POINT 4
YCOMP. POINT 4

FIGURE T17

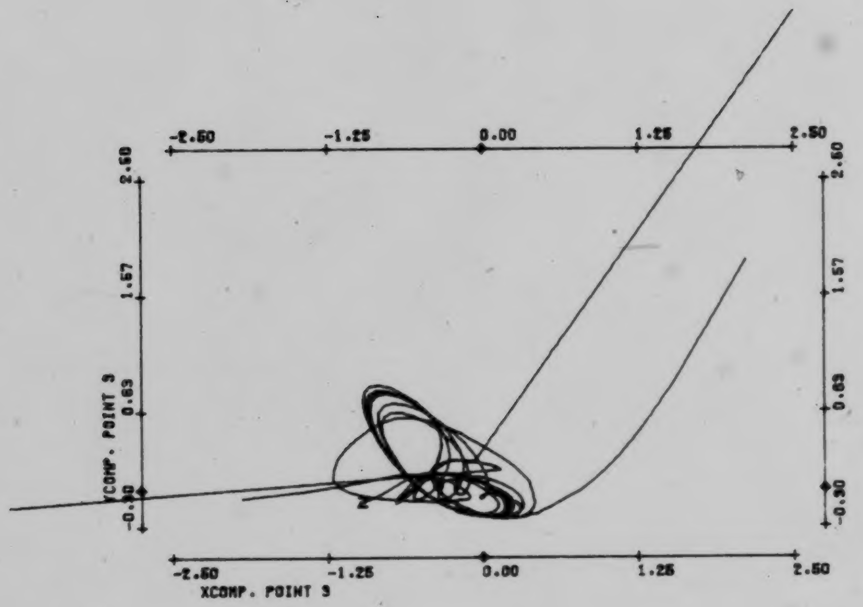
Bifurcation parameter η	.445
No. of timesteps	500
Stepsize	.03
Init. condition for $\vec{V}_2$	.1 (all)
Average size of spatial random perturbations	.01
Average no. of timesteps between random perturbations	2
Period of $\vec{V}_1$	1.55
Mean velocity:	
Point 1	1.61
2	1.68
3	2.67
4	1.76

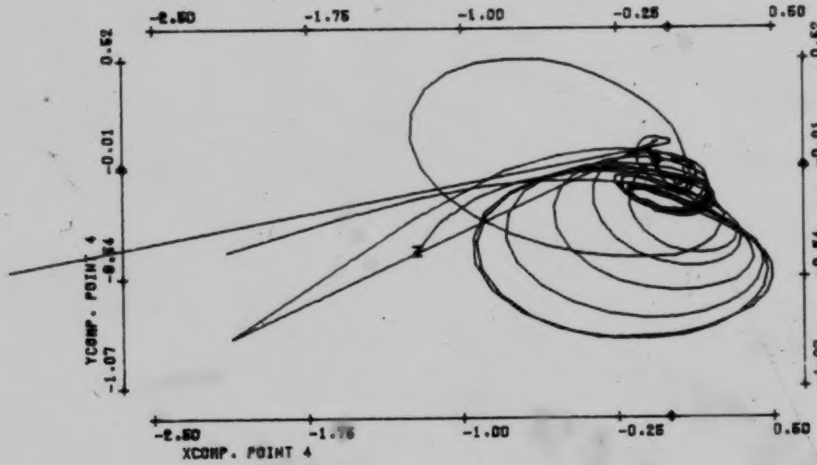
T17(1)

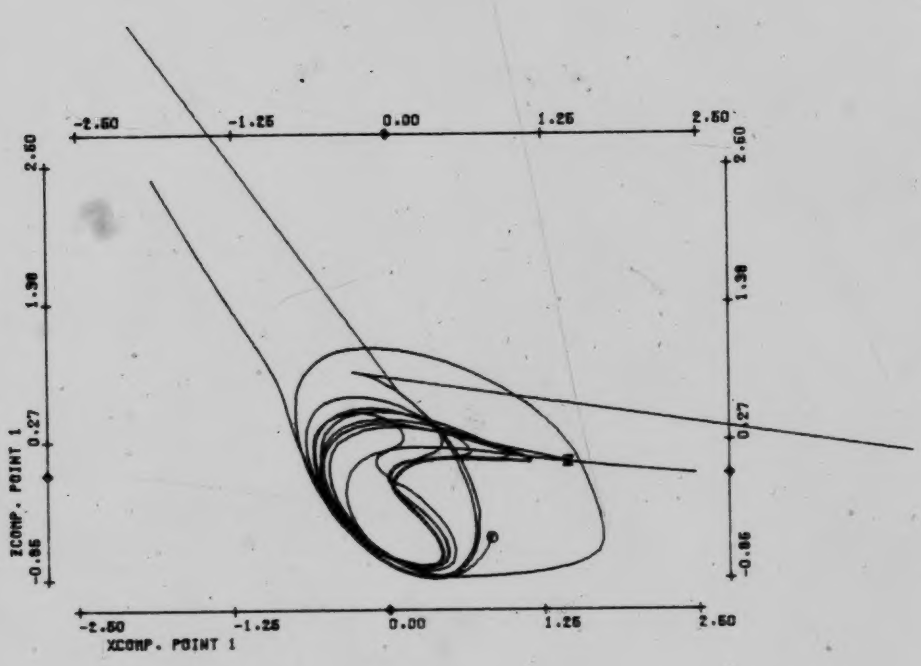

-2.50
-1.25
0.00
1.25
2.50
0.63
-0.48
-1.48
-2.60
Z
YCOMP. POINT 1
XCOMP. POINT 1

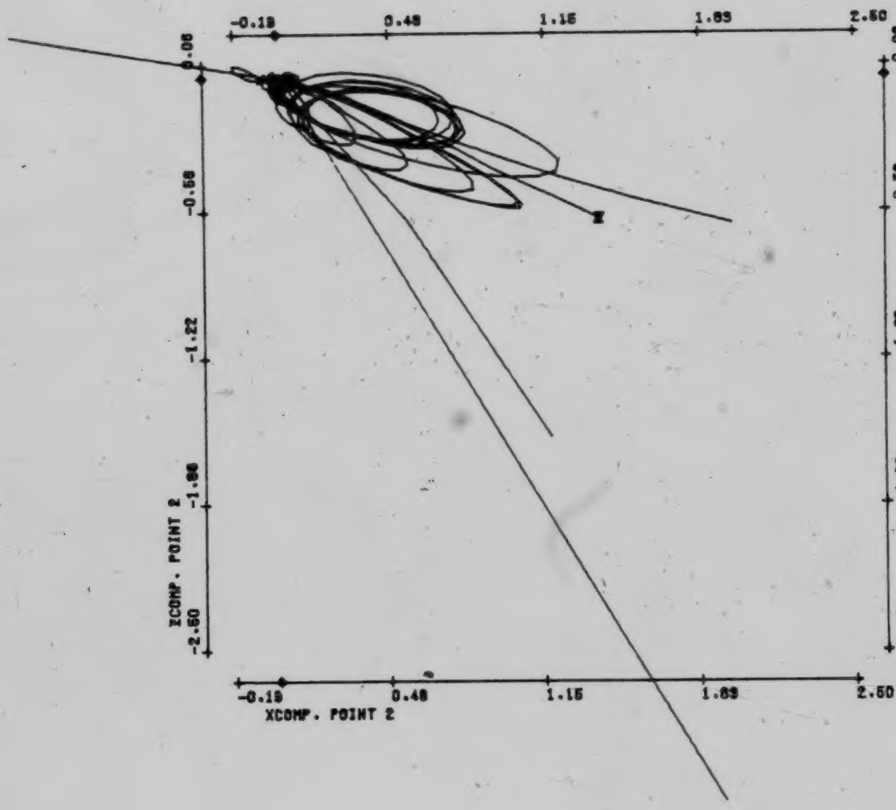

-0.19
0.48
1.15
1.83
2.50
1.70
0.90
0.10
-0.70
Z
YCOMP. POINT 2
XCOMP. POINT 2

T17(2)

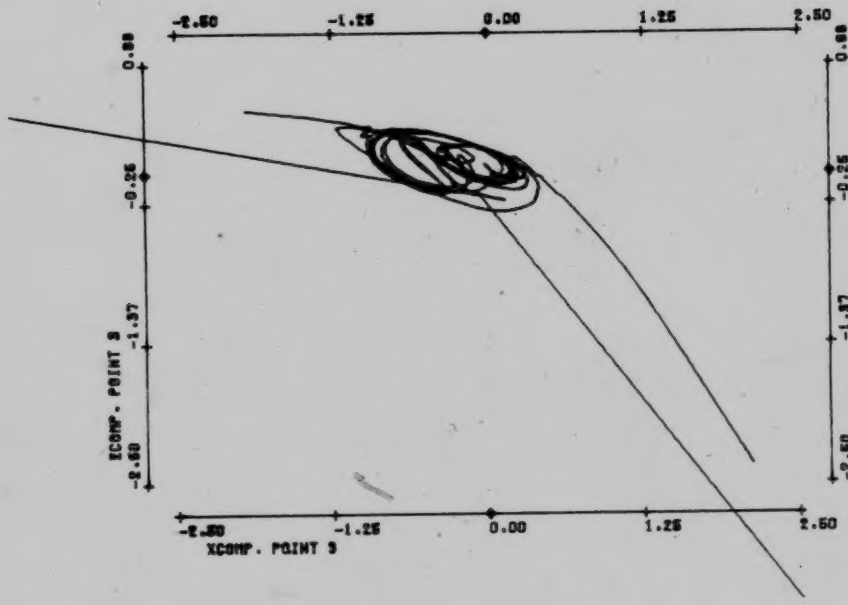

-2.50
-1.25
0.00
1.25
2.50
2.50
1.57
0.63
-0.30
YCOMP. POINT 3
XCOMP. POINT 3
Z

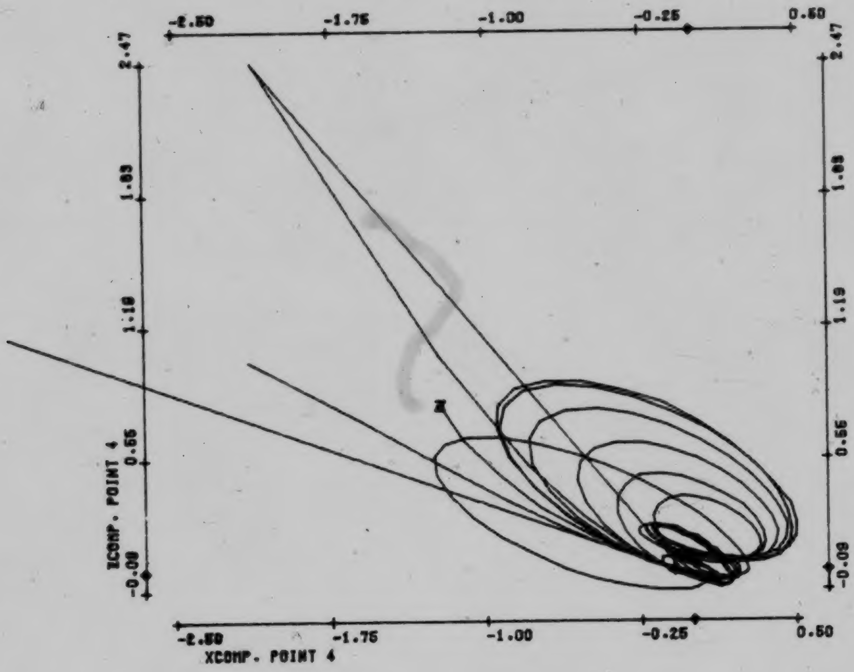

-2.50
-1.75
-1.00
-0.25
0.50
0.52
-0.01
-0.54
-1.07
YCOMP. POINT 4
XCOMP. POINT 4
Z


-2.50
-1.25
0.00
1.25
2.50
2.60
1.38
0.27
-0.86
ZCOMP. POINT 1
XCOMP. POINT 1

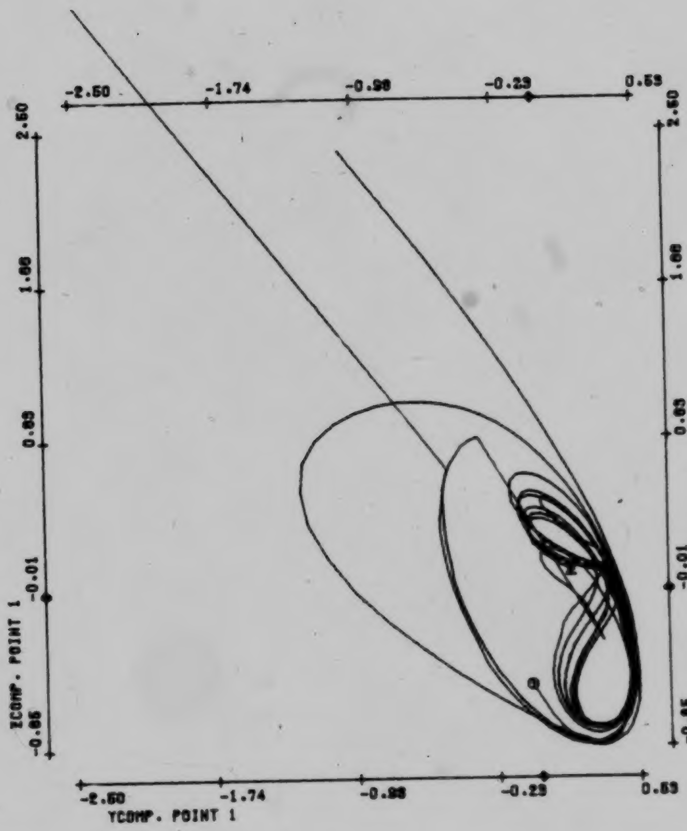

-0.19
0.48
1.16
1.83
2.50
0.06
-0.58
-1.22
-1.86
-2.50
ZCOMP. POINT 2
XCOMP. POINT 2

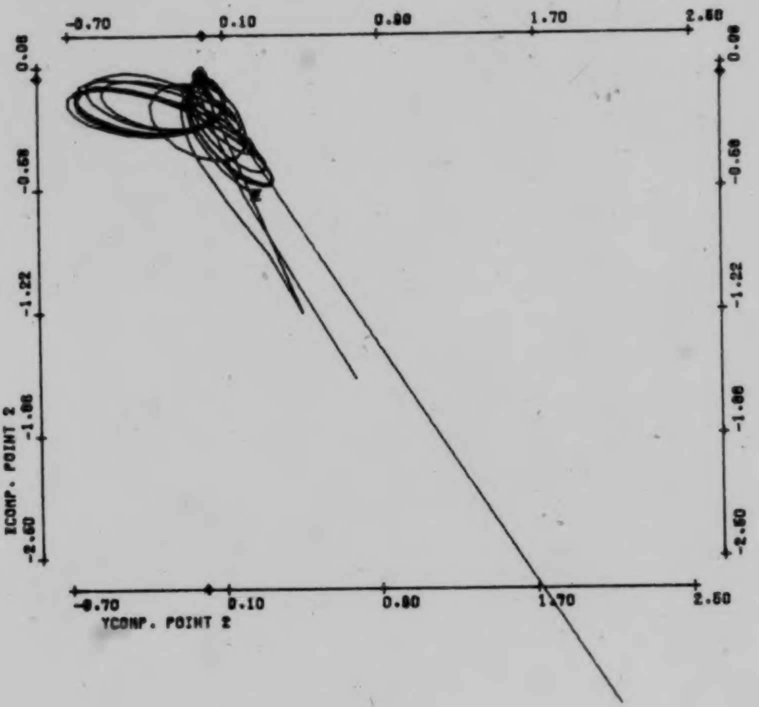
T17(4)

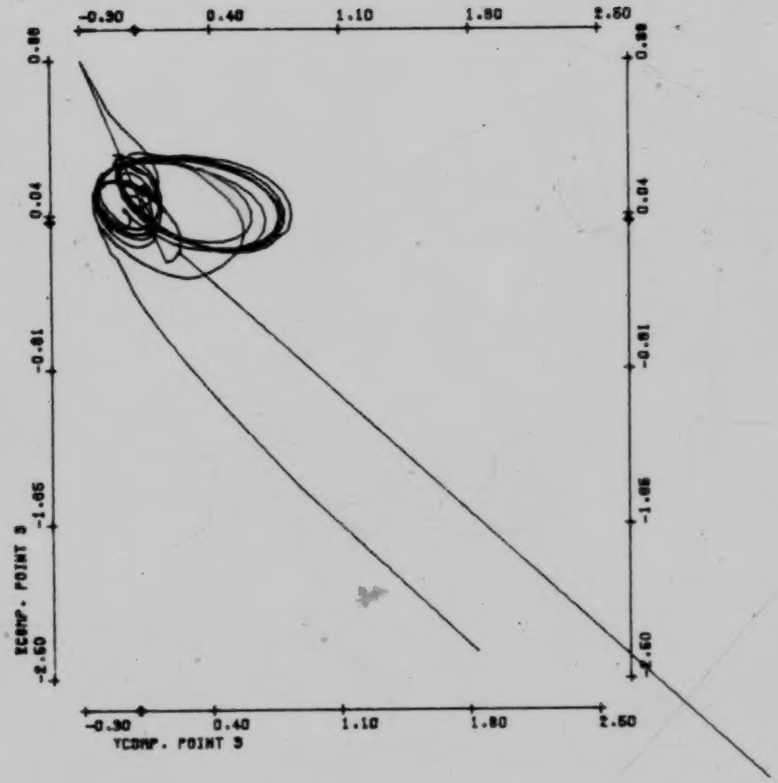

-2.50
-1.25
0.00
1.25
2.50
0.88
-0.25
-1.37
-2.50
ZCOMP. POINT 3
XCOMP. POINT 3


-2.50
-1.75
-1.00
-0.25
0.50
2.47
1.83
1.19
0.55
-0.09
ZCOMP. POINT 4
XCOMP. POINT 4

T17(5)


-2.50
-1.74
-0.98
-0.23
0.53
2.50
1.66
0.83
-0.01
-0.85
ZCOMP. POINT 1
YCOMP. POINT 1
1
0


-0.70
0.10
0.90
1.70
2.50
0.06
-0.58
-1.22
-1.86
-2.50
ZCOMP. POINT 2
YCOMP. POINT 2
2


-0.30
0.40
1.10
1.80
2.50
0.88
0.04
-0.81
-1.65
-2.50
ZCOMP. POINT 3
YCOMP. POINT 3

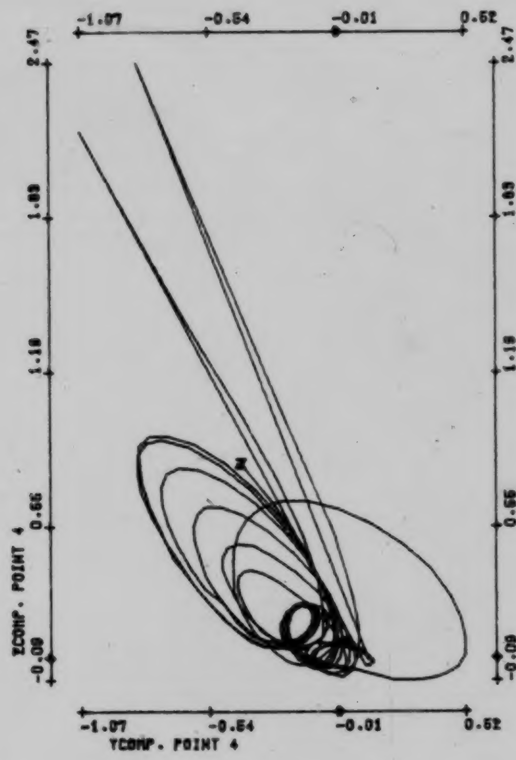
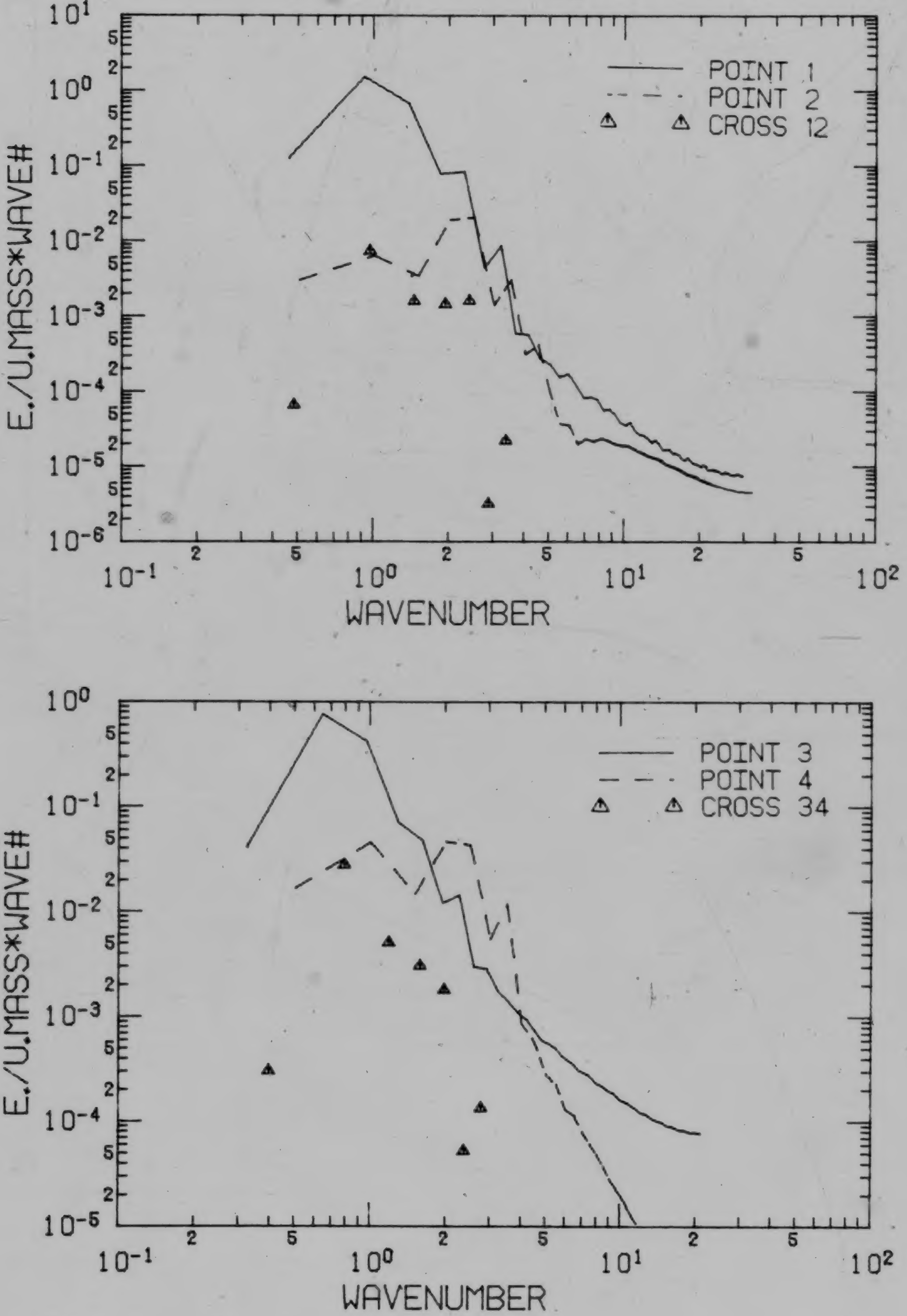

-1.07
-0.54
-0.01
0.52
2.47
1.83
1.18
0.55
-0.09
ZCOMP. POINT 4
YCOMP. POINT 4

FIGURE S3

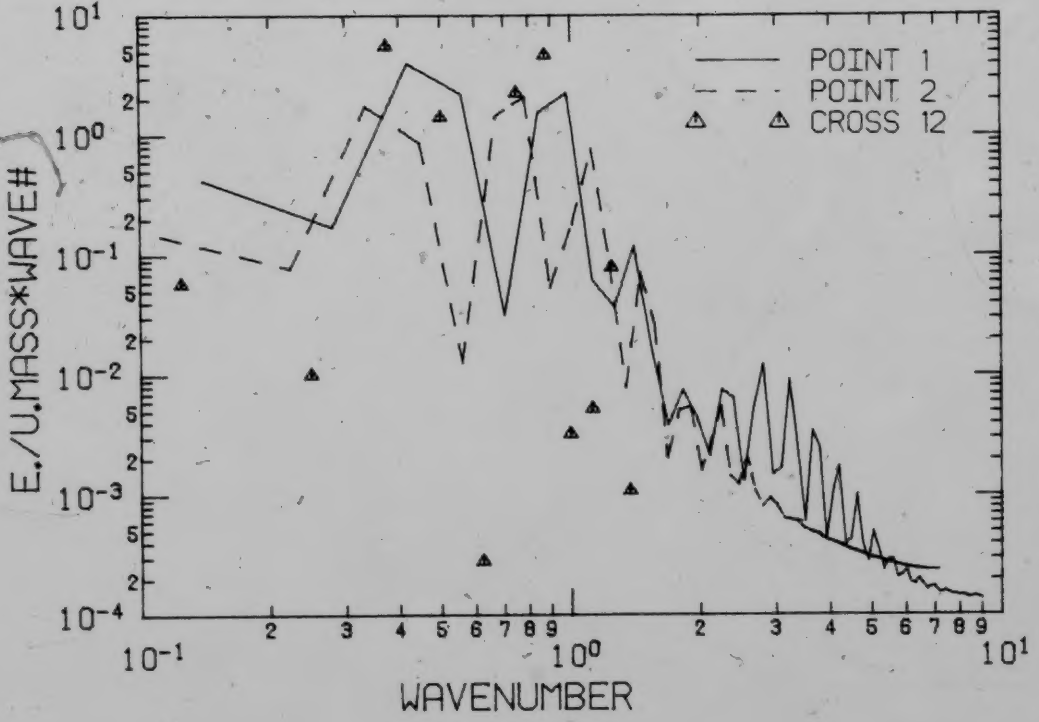
Bifurcation parameter η	.3
No. of timesteps	1000
Stepsize	.03
Init. condition for $\vec{V}_2$	.001 (all)
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.68
Mean velocity	
Point 1	2.20
2	2.03
3	2.95
4	1.85
Sample size/subsample size = no. of subsamples	1024/128 =8

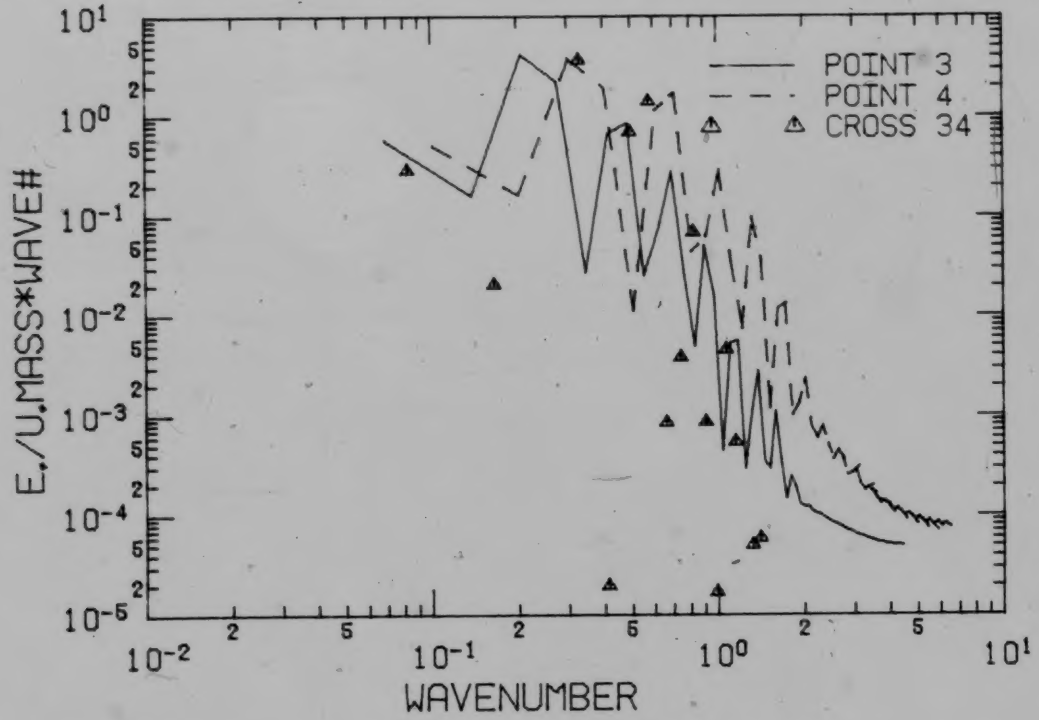


ENERGY SPECTRUM OF PERTURBATIONS

FIGURE S4

Bifurcation parameter η	.449
No. of timesteps	800
Stepsize	.04
Init. condition for $\vec{V}_2$	.001(X-comp.Pt.1) rest - 0
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.55
Mean velocity:	
Point 1	1.38
2	1.76
3	2.84
4	1.95
Sample size/subsample size = no. of subsamples	896/128 = 7

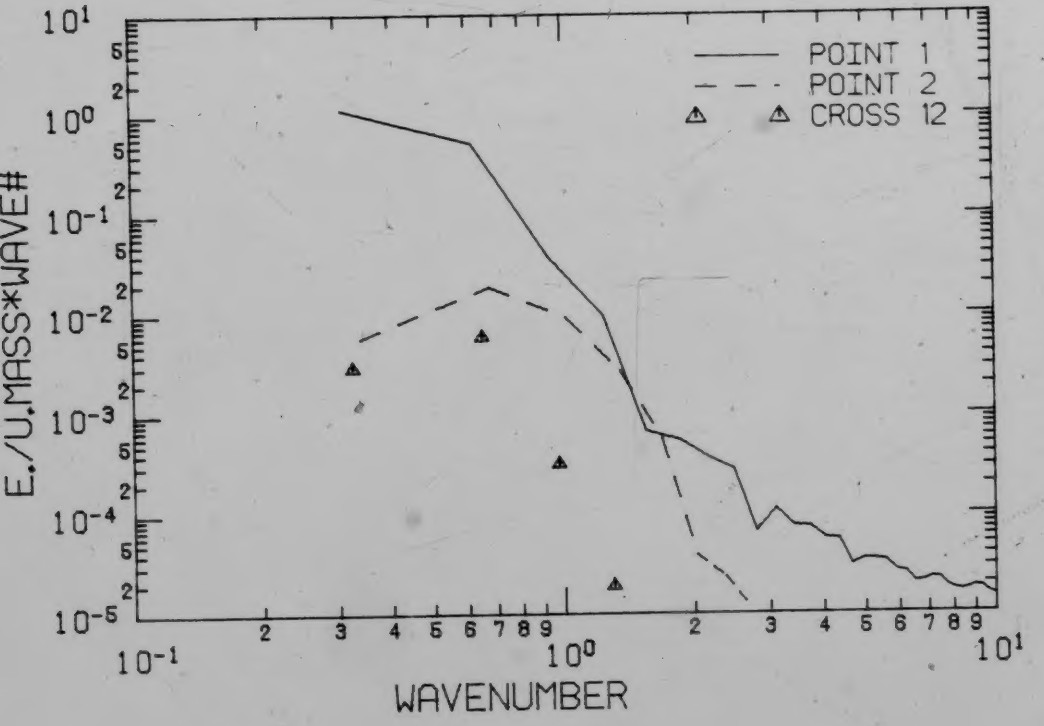


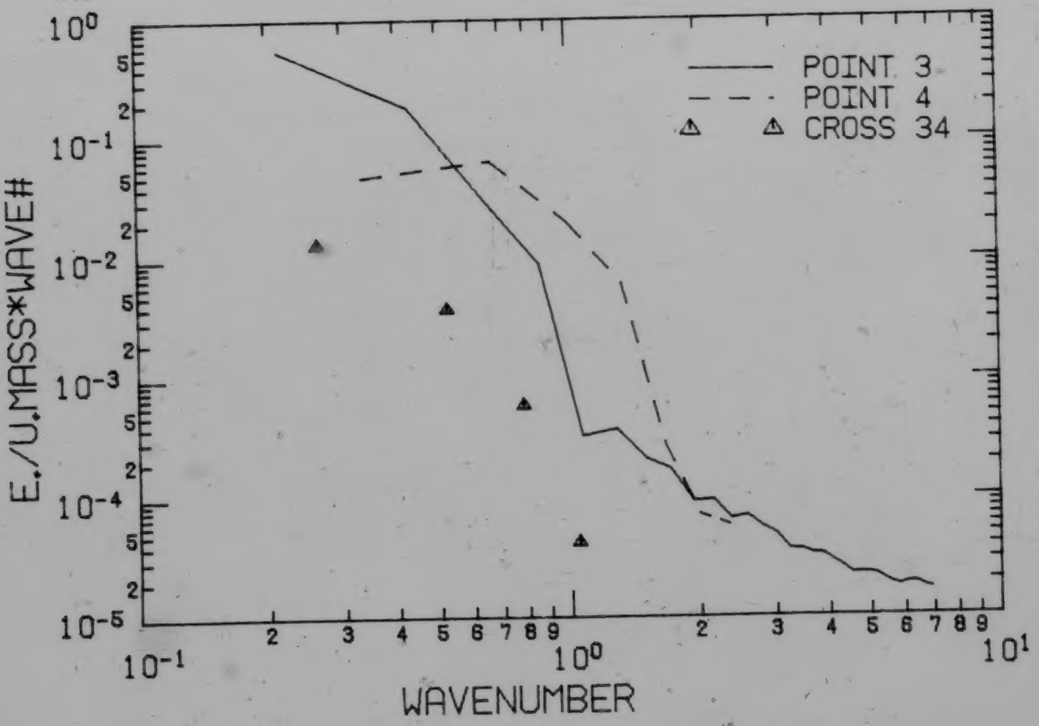


ENERGY SPECTRUM OF PERTURBATIONS

FIGURE S5

Bifurcation parameter η	.325
No. of timesteps	500
Stepsize	.03
Init. condition for $\vec{V}_2$	.001 (all)
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.66
Mean velocity:	
Point 1	1.95
2	1.64
3	2.65
4	1.60
Sample size/subsample size = no. of subsamples	512/64 = 8

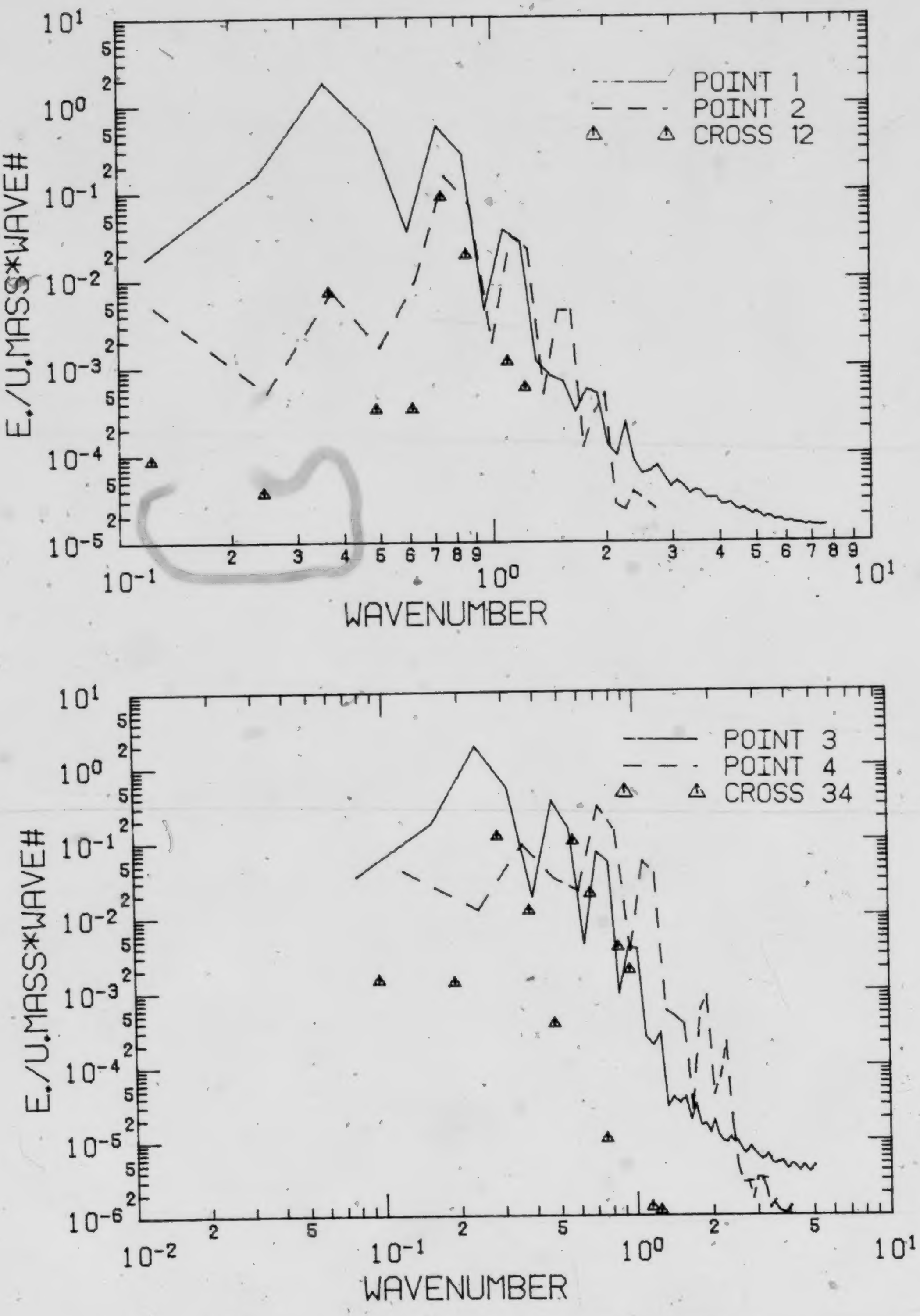




ENERGY SPECTRUM OF PERTURBATIONS

FIGURE S6

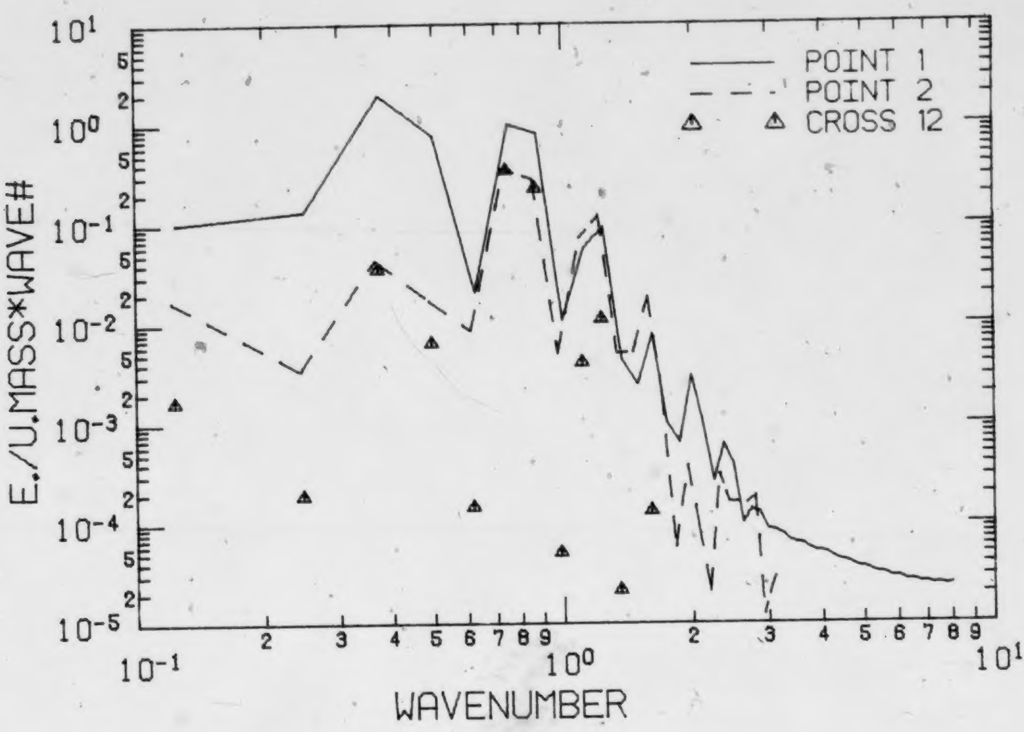
Bifurcation parameter η	.35
No. of timesteps	1000
Stepsize	.04
Init. condition for $\vec{V}_2$	.001 (X-comp. Pt.1) rest = 0
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.63
Mean velocity:	
Point 1	
2	
3	
4	
Sample size/subsample size = no. of subsamples	1024/128 = 8

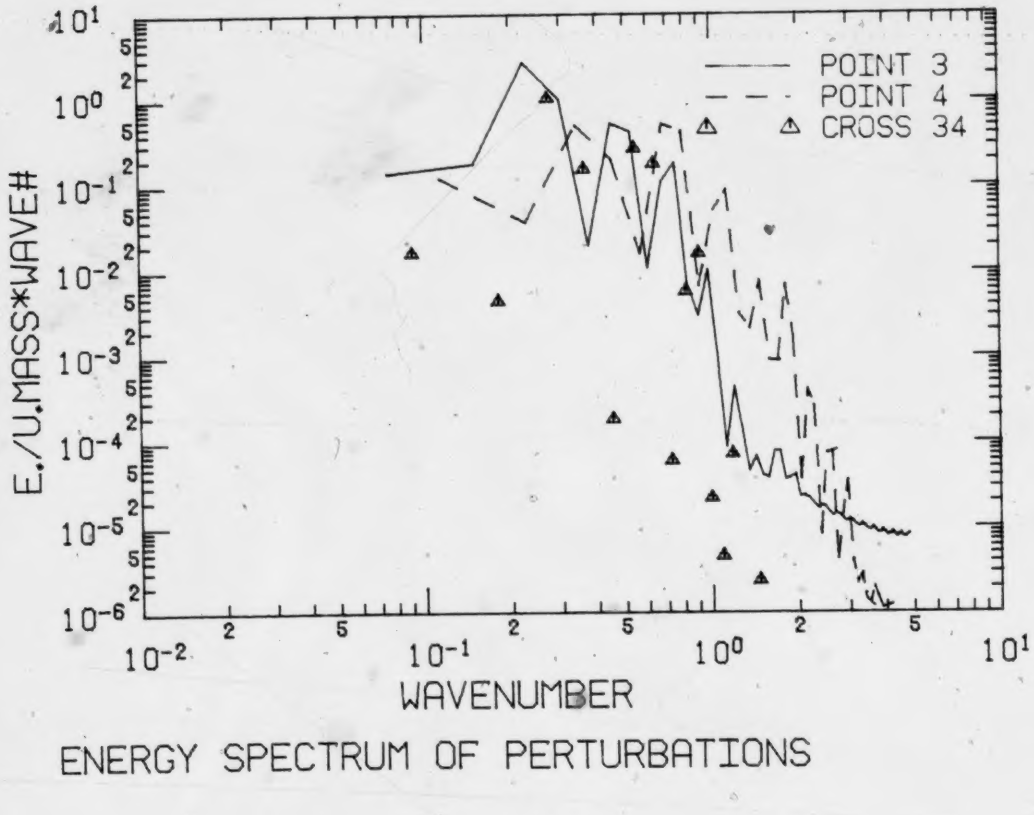


ENERGY SPECTRUM OF PERTURBATIONS

FIGURE S7

Bifurcation parameter η	.4
No. of timesteps	800
Stepsize	.04
Init. condition for $\vec{V}_2$	.001 (X-comp.Pt.1) rest = 0
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.60
Mean velocity:	
Point 1	1.56
2	1.61
3	2.61
4	1.73
Sample size/subsample size = no. of subsamples	896/128 = 7

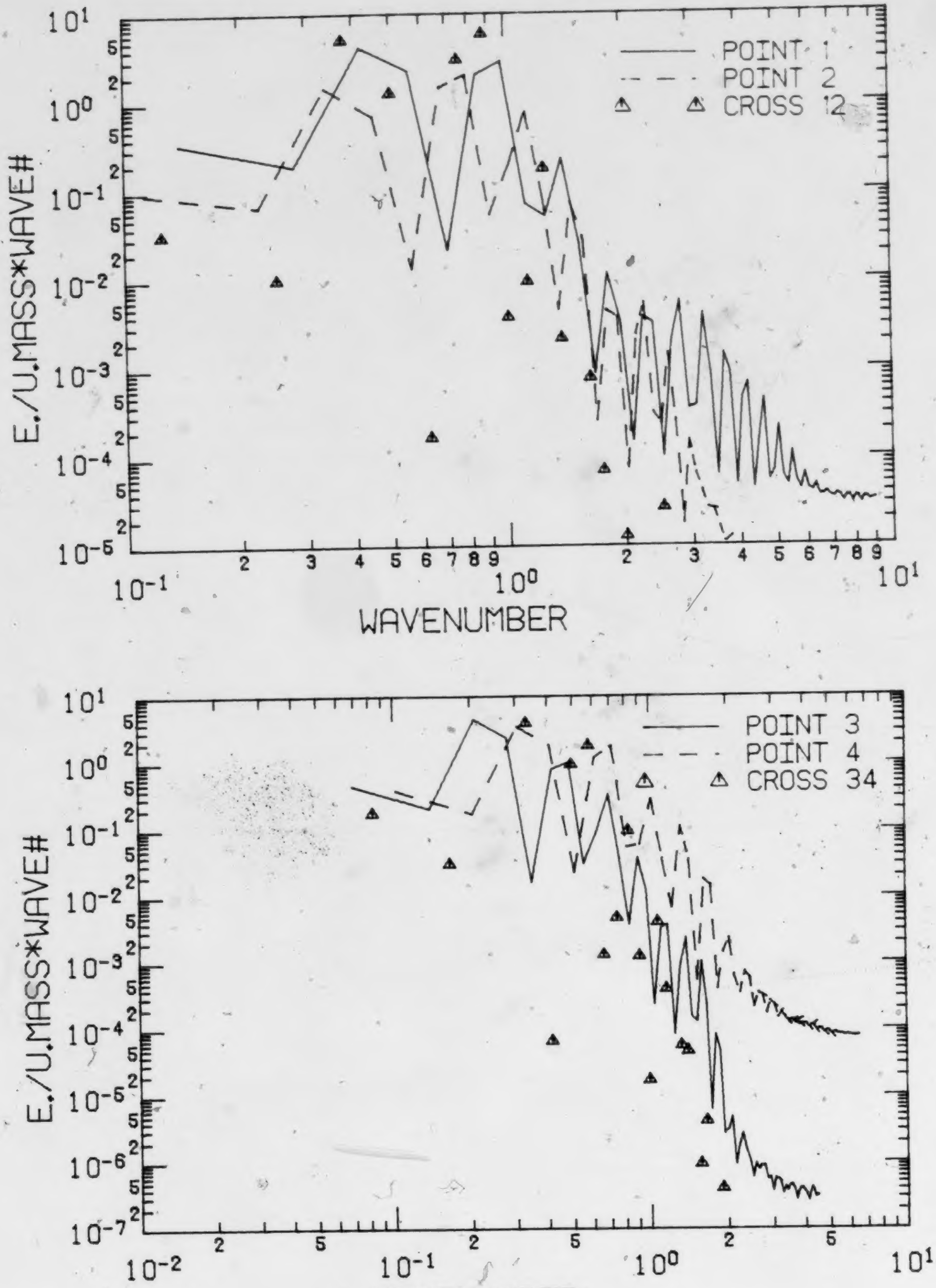




ENERGY SPECTRUM OF PERTURBATIONS

FIGURE S8

Bifurcation parameter η	.447
No. of timesteps	1000
Stepsize	.04
Init. condition for $\vec{V}_2$	.001(X-comp.Pt.1) rest = 0
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.55
Mean velocity:	
Point 1	1.39
2	1.73
3	2.81
4	1.91
Sample size/subsample size = no. of subsamples	1024/128 = 8

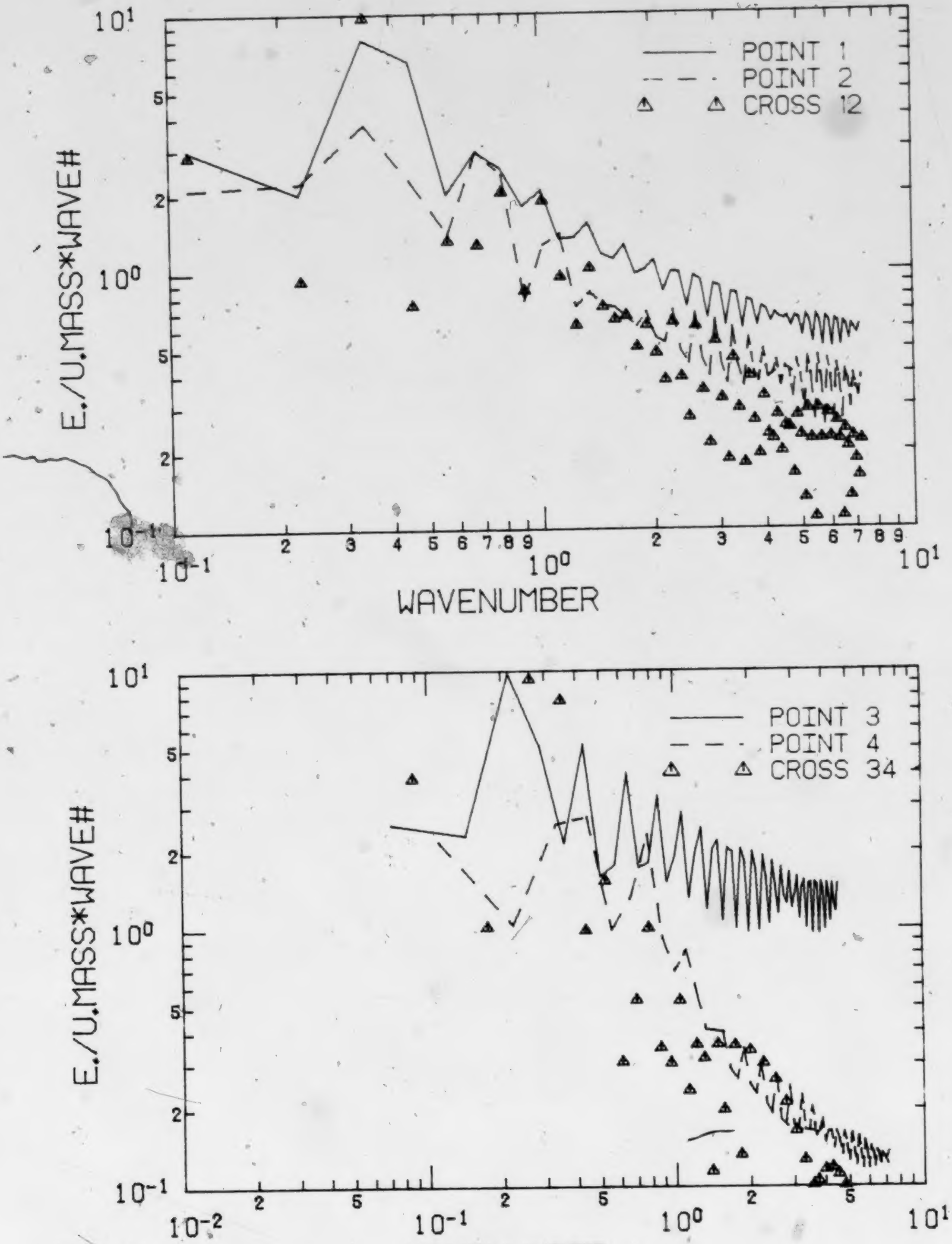


ENERGY SPECTRUM OF PERTURBATIONS

FIGURE S9

Bifurcation parameter η	.47
No. of timesteps	1000
Stepsize	.04
Init. condition for $\vec{V}_2$	.01(X-comp.Pt.1) rest = 0
Average size of spatial random perturbations	n/a
Average no. of timesteps between random perturbations	n/a
Period of $\vec{V}_1$	1.53
Mean velocity:	
Point 1	1.71
2	1.76
3	2.78
4	1.83
Sample size/subsample size = no. of subsamples	1024/128 = 8

S9

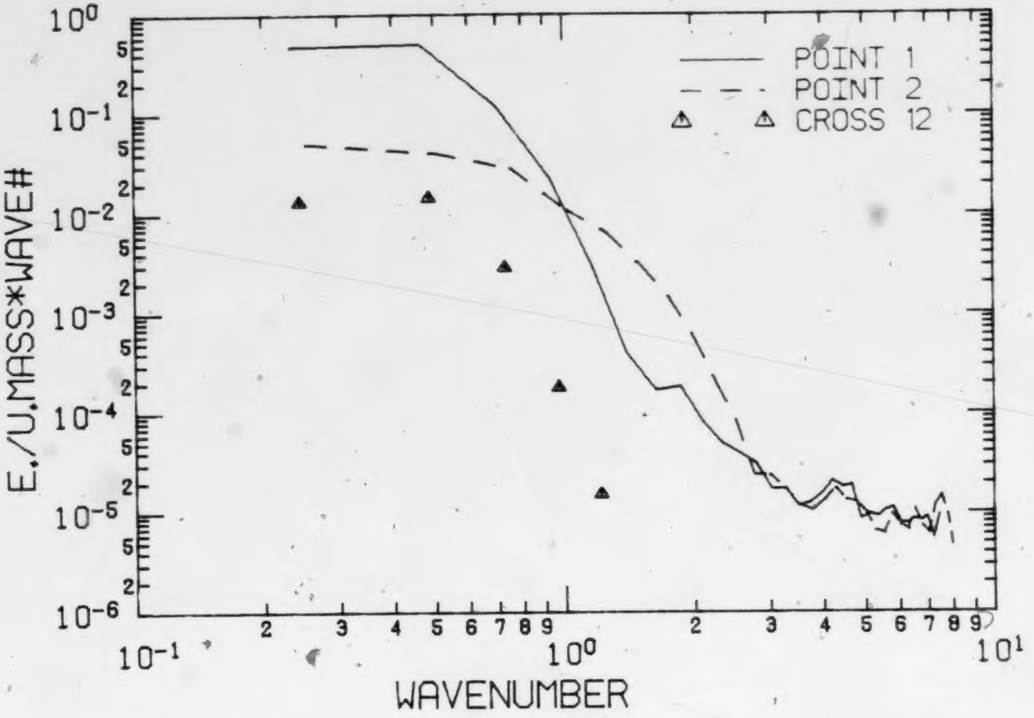


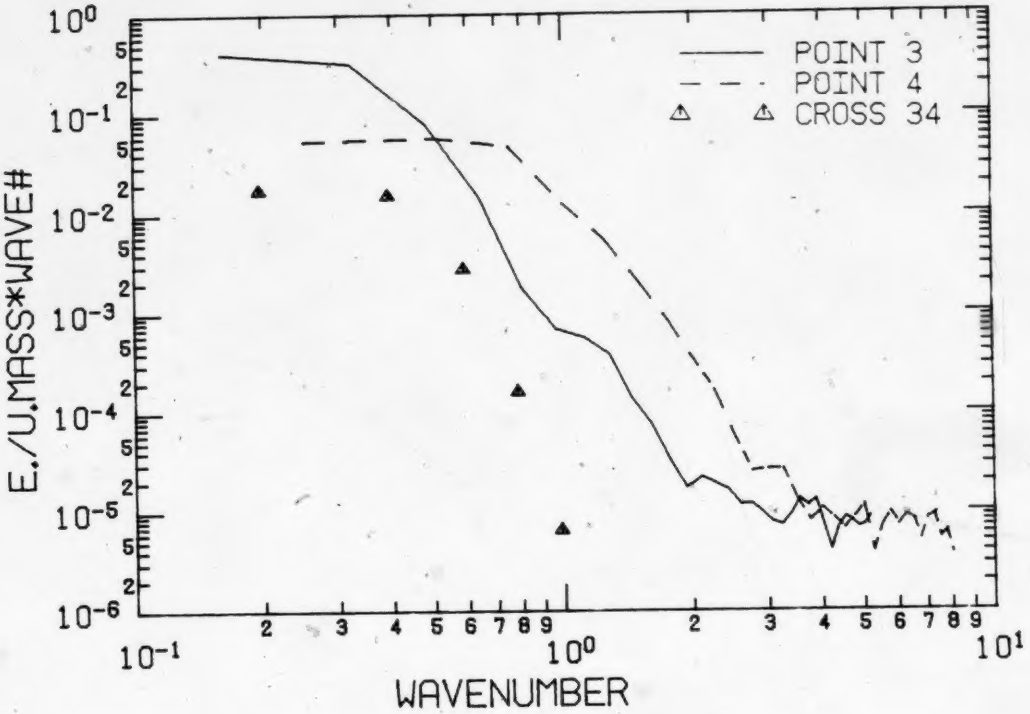
ENERGY SPECTRUM OF PERTURBATIONS

FIGURE S12

Bifurcation parameter η	.27
No. of timesteps	1000
Stepsize	.04
Init. condition for $\vec{V}_2$	.1 (all)
Average size of spatial random perturbations	.01
Average no. of timesteps between random perturbations	1.5
Period of $\vec{V}_1$	1.7
Mean velocity:	
Point 1	1.68
2	1.56
3	2.43
4	1.56
Sample size/subsample size = no. of subsamples	768/64 = 12

(First 233 timesteps are unperturbed.)





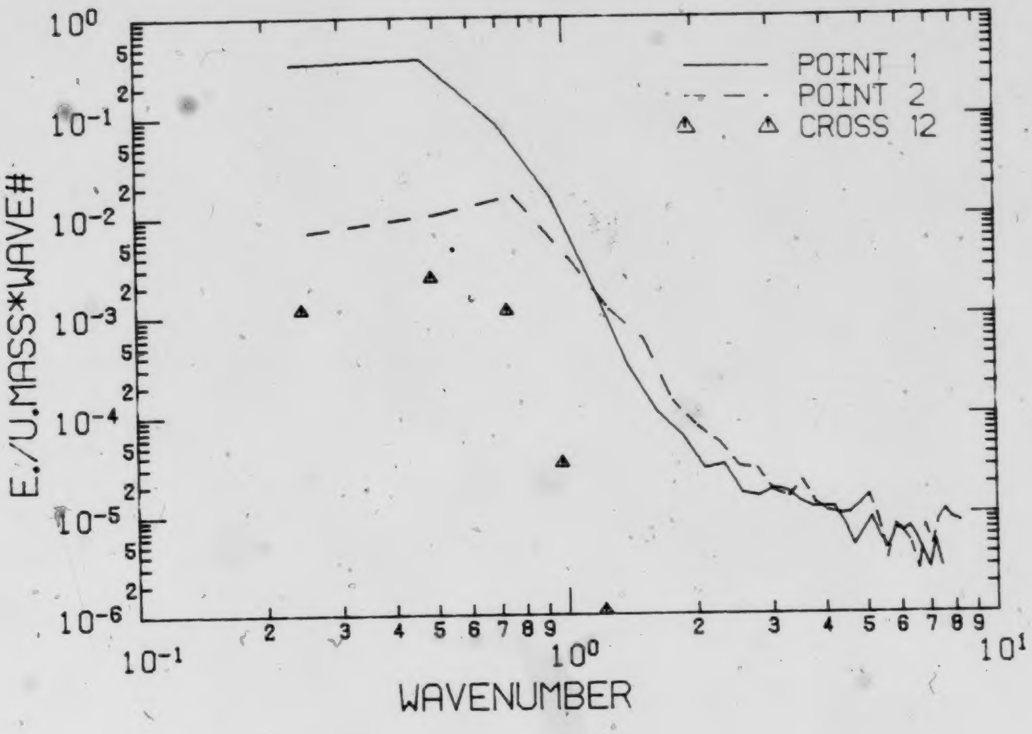
ENERGY SPECTRUM OF PERTURBATIONS

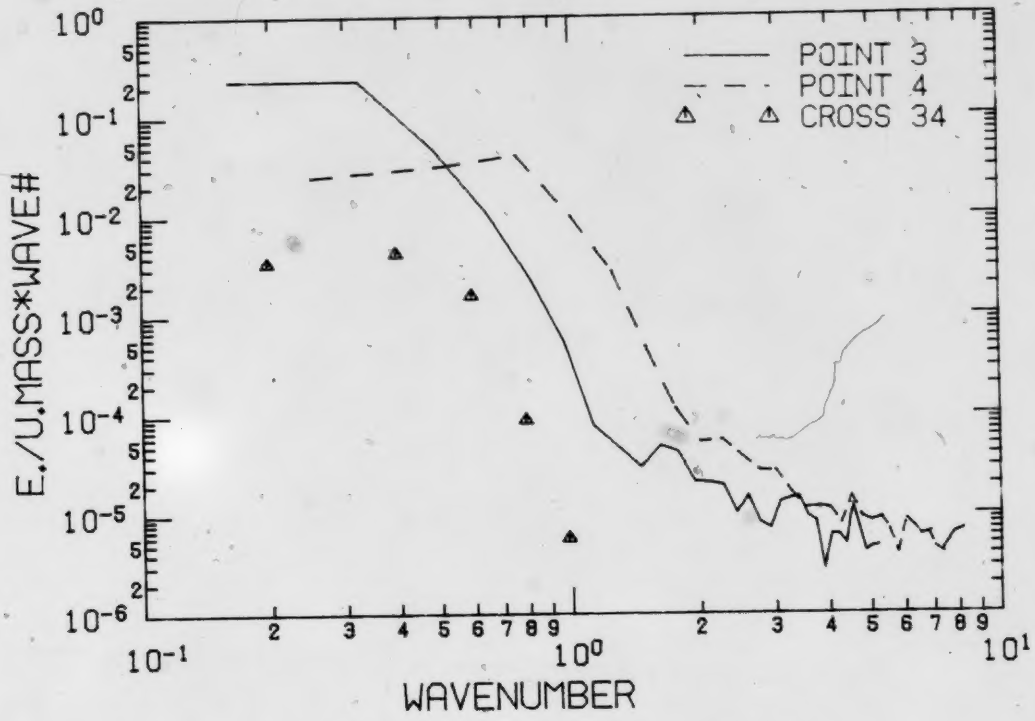
FIGURE S13

Bifurcation parameter η	.28
No. of timesteps	1000
Stepsize	.04
Init. condition for $\vec{V}_2$	.01 (all)
Average size of spatial random perturbations	.01
Average no. of timesteps between random perturbations	2
Period of $\vec{V}_1$	1.69
Mean velocity:	
Point 1	1.68
2	1.55
3	2.42
4	1.56
Sample size/subsample size = no. of subsamples	768/64 = 12

(First 233 timesteps are unperturbed.)

S13

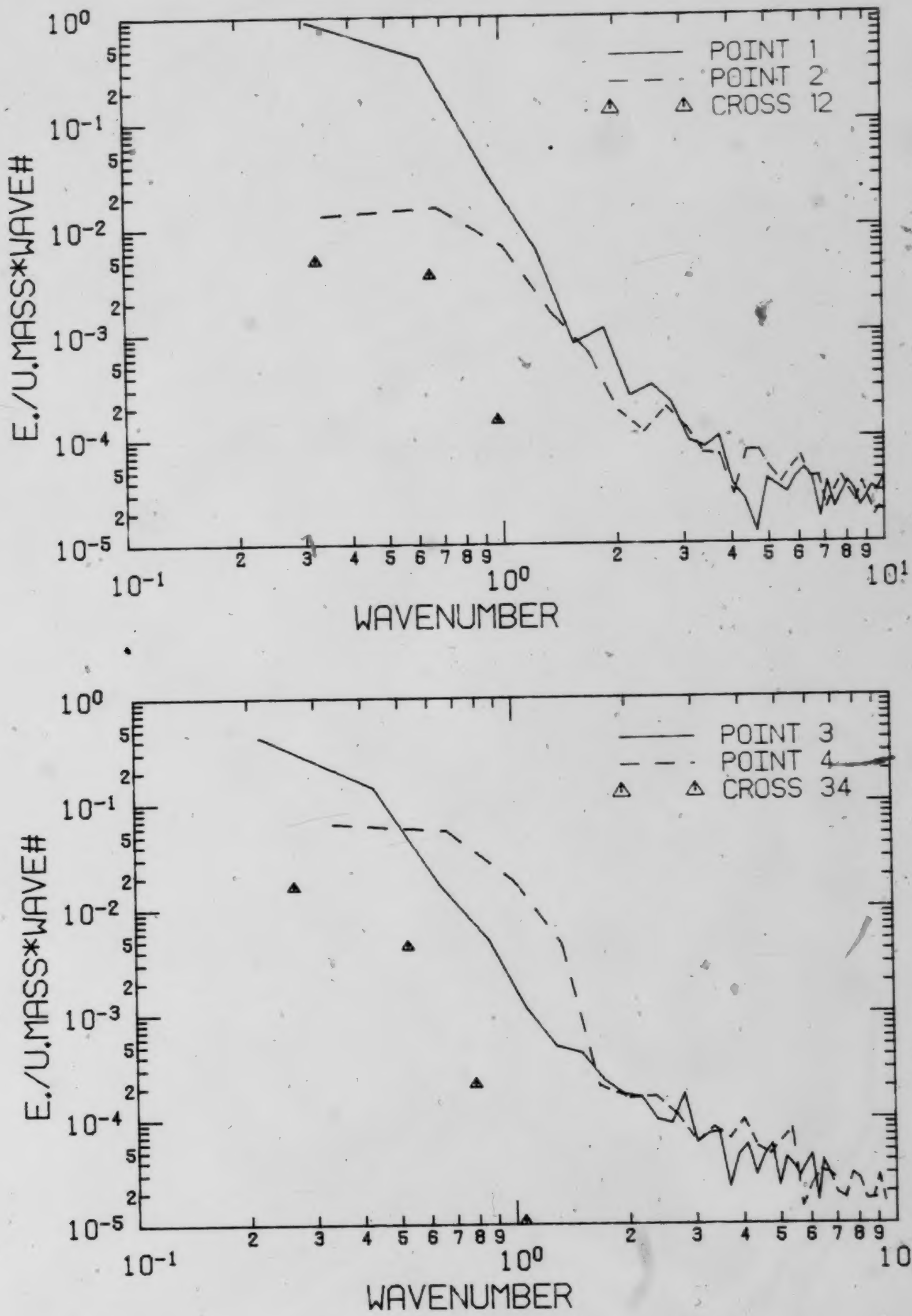




ENERGY SPECTRUM OF PERTURBATIONS

FIGURE S14

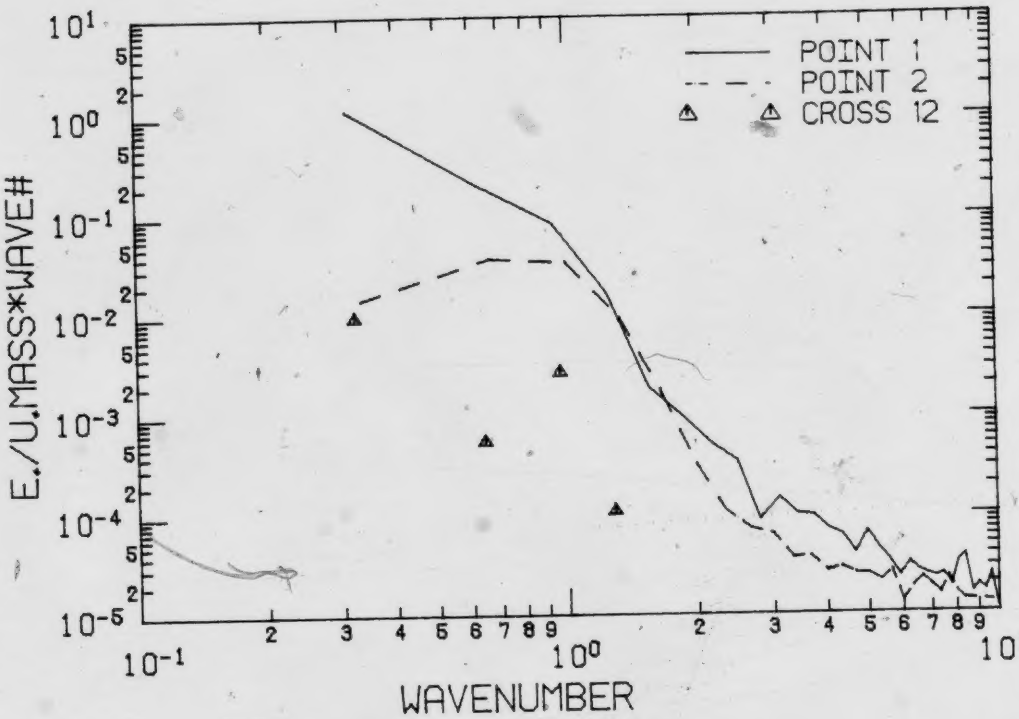
Bifurcation parameter η	.3
No. of timesteps	500
Stepsize	.03
Init. condition for $\vec{V}_2$	0 (all)
Average size of spatial random perturbations	.02
Average no. of timesteps between random perturbations	2
Period of $\vec{V}_1$	1.68
Mean velocity:	
Point 1	2.20
2	2.03
3	2.95
4	1.85
Sample size/subsample size = no. of subsamples	512/64 = 8

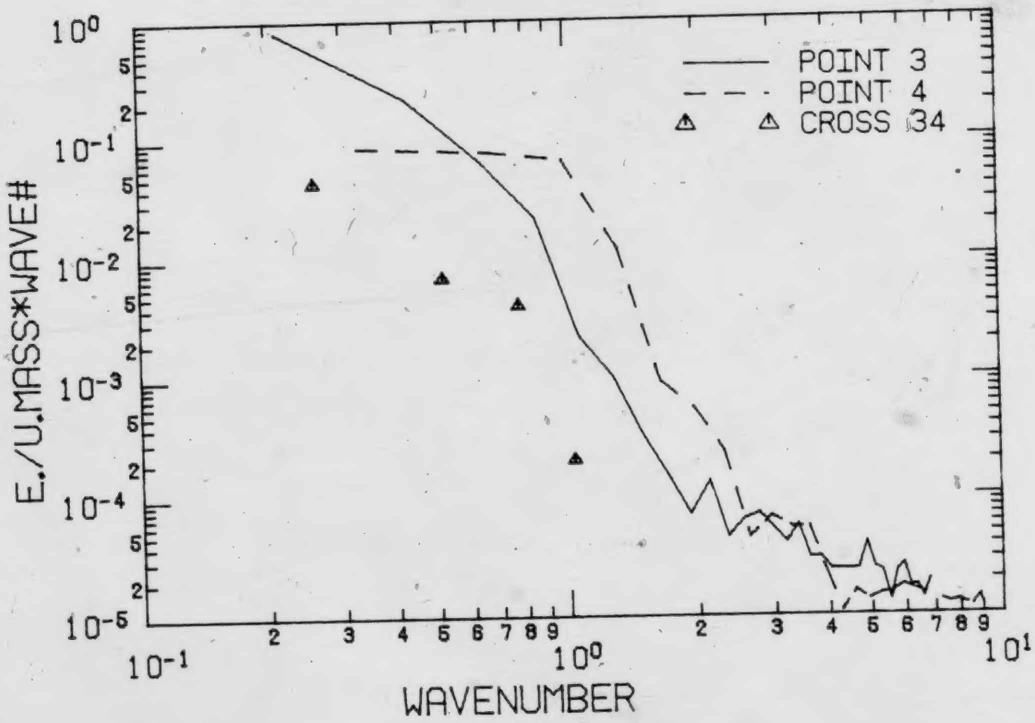


ENERGY SPECTRUM OF PERTURBATIONS

FIGURE S15

Bifurcation parameter η	.325
No. of timesteps	500
Stepsize	.03
Init. condition for $\vec{V}_2$	0 (all)
Average size of spatial random perturbations	.015
Average no. of timesteps between random perturbations	2.5
Period of $\vec{V}_1$	1.66
Mean velocity:	
Point 1	1.95
2	1.64
3	2.65
4	1.60
Sample size/subsample size = no. of subsamples	512/64 = 8



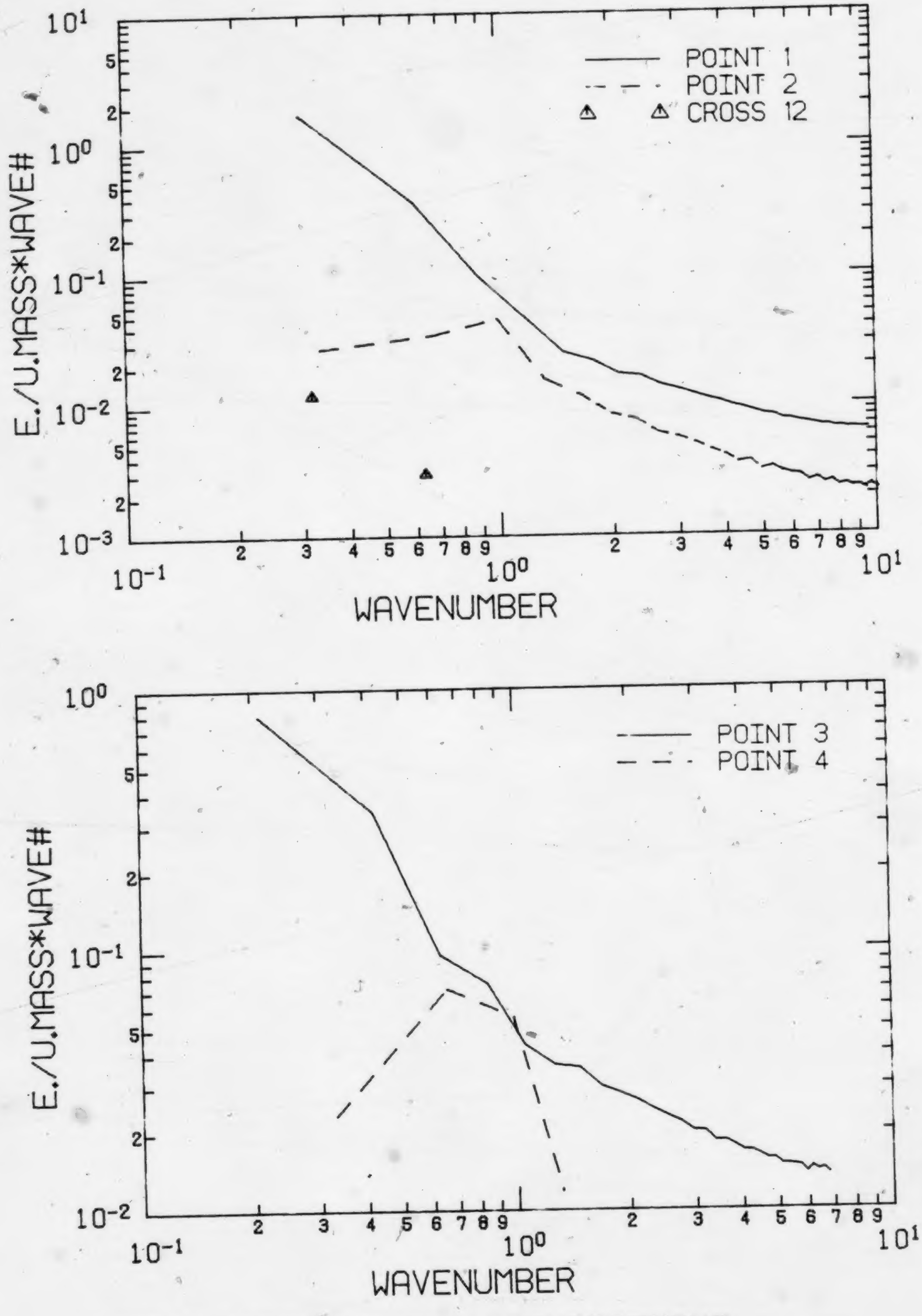


ENERGY SPECTRUM OF PERTURBATIONS

FIGURE S16

Bifurcation parameter η	.33
No. of timesteps	645
Stepsize	.03
Init. condition for $\vec{V}_2$	.001(X-comp.Pt.1) rest = 0
Average size of spatial random perturbations	.01
Average no. of timesteps between random perturbations	2
Period of $\vec{V}_1$	1.65
Mean velocity:	
Point 1	1.75
2	1.58
3	2.43
4	1.59
Sample size/subsample size = no. of subsamples	448/64 = 7

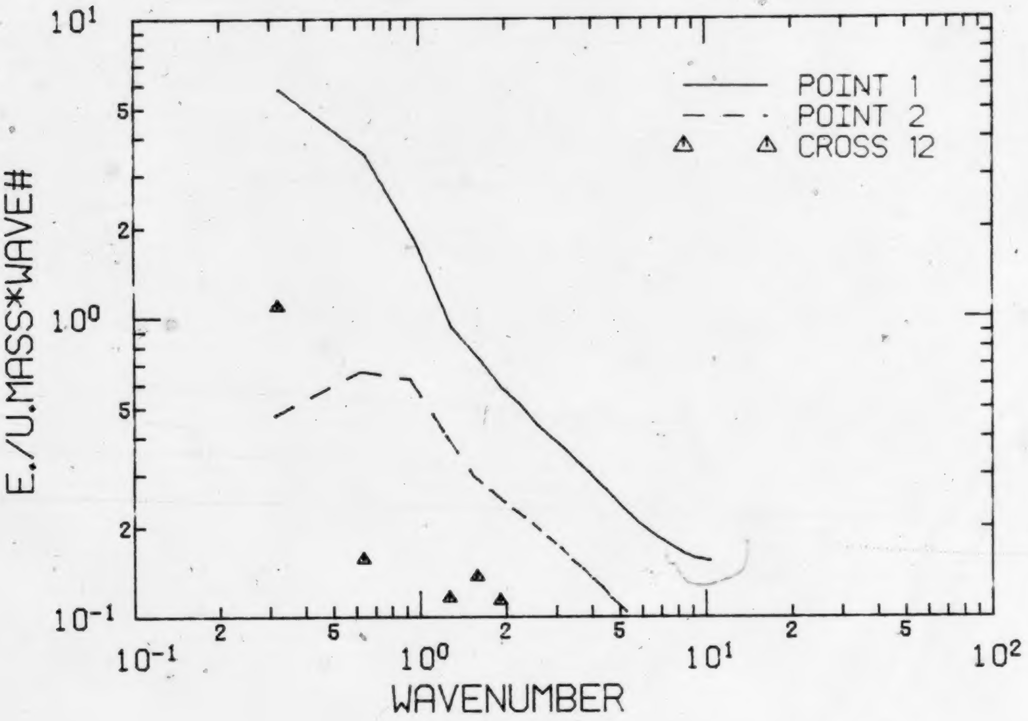
(First 200 timesteps are unperturbed.)

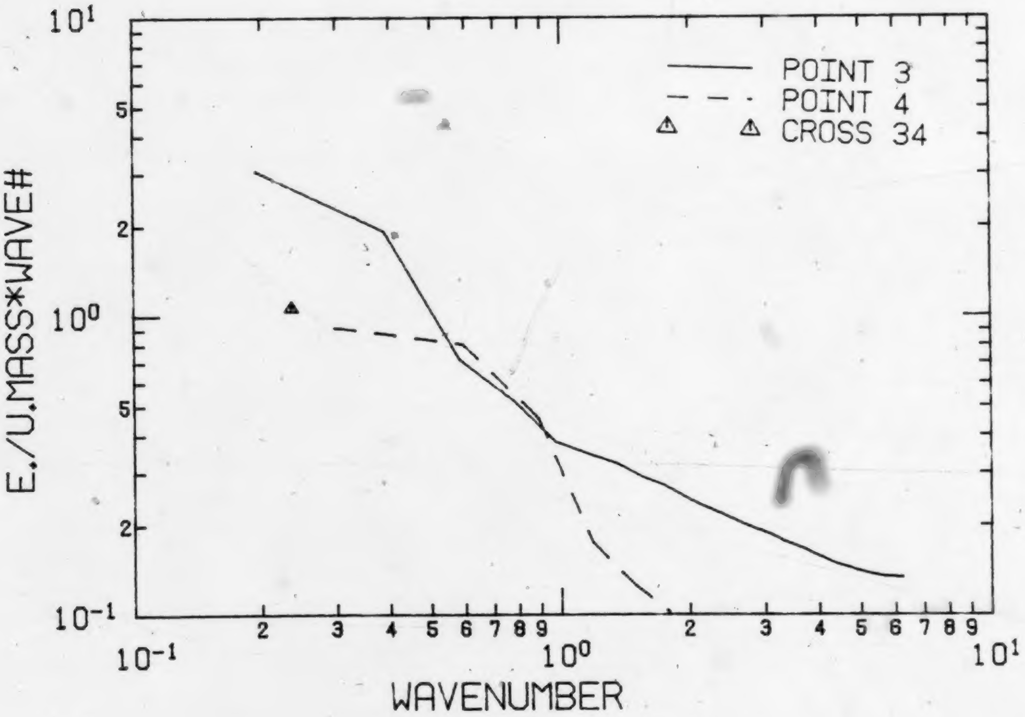


ENERGY SPECTRUM OF PERTURBATIONS

FIGURE S17

Bifurcation parameter η	.445
No. of timesteps	500
Stepsize	.03
Init. condition for $\vec{V}_2$	.1 (all)
Average size of spatial random perturbations	.01
Average no. of timesteps between random perturbations	2
Period of $\vec{V}_1$	1.55
Mean velocity:	
Point 1	1.61
2	1.68
3	2.67
4	1.76
Sample size/subsample size = no. of subsamples	512/64 = 8





ENERGY SPECTRUM OF PERTURBATIONS

APPENDIX G

1. PROGRAM LISTINGS

The following programs and files are listed:

1) DSCRB (+ subroutines)
2) ADV
3) BIF3A (+ subroutines FUN, FUNP, MFC)
4) ODE.M (+ subroutine F45)
5) Control parameter file 4.NS.A.

DSCRB with subroutines and ADV are described in Appendix B. BIF3A is the calling program for the subroutine BIFOR 2 (see Chapter 8). BIFOR 2 is not listed here,* since it could be used in its original form with only the following modifications for the present study:

1) Conversion to double precision.

   This was done by W.G. Aiello, Department of Mathematics, University of Alberta. It was necessary since BIFOR 2 was written on a CDC 6400, which is in its single-precision mode almost as accurate as an AMDAHL (computer at U. of A.) in double precision mode. Therefore, algorithms accumulated roundoff errors too fast and could not meet their tolerances on AMDAHL single precision mode. This was the only major modification.

2) The switch ICK was added (via parameter string) to skip the routine CHECKJ on option, which evaluates the Jacobian of the system by

---

*A listing is available as microfiche in [5] or directly through its author: B. Hassard, Dept. of Mathematics, SUNY, Buffalo, NY.

finite differencing and compares it against an analytic evaluation done by the user-supplied subroutine FUN (or FUNP respectively). This is just a CPU-time saving measure, since this test has to be performed only once to insure a correct analytic evaluation of the Jacobian by FUN.

3) The parameters EPSR (tolerance for the above mentioned comparison test and zero-criterion for the eigenvalue of the Jacobian), EPS, NSIG (parameters used for zero-criteria in various algorithms and iterations used) and ITMAX (maximum number of iterations allowed in various algorithms) can be set externally through parameter string.

4) The microblock COMMON/CT/ICT was introduced to BIFOR2 and FUN. The switch ICT is reset from zero to one after the critical value of the bifurcation parameter has been found and is used in FUN to save a copy of the coefficient matrix of the system and its transpose at that point for evaluation of the mean flow in MFC and possibly the second and third level bifurcations in FUNP.

A peculiarity, which is MTS*-specific, is the requirement to multiply the line number in sequential READ and WRITE statements by a factor of 1000. This can be seen in the transposing sections of BIF3A and ADV and in all READ statements from logical unit 4 (attached to control parameter file 4.NS.A).

To set up a particular discretization and flow profile, one would proceed as follows:

---

[*] Michigan Terminal System: Operating system used on the University of Alberta, AMDAHL computer.

1) Set maximum number of increments in X-, Y-, Z-direction on line 136 of 4.NS.A. The numbers chosen are equal to the numbers of corners of rectangular blocks and equal the numbers of blocks minus one respectively in the particular directions (maximum number of increments is 5).

2) Set corner coordinates of blocks in X-, Y-, Z-direction in lines 138 to 140 of 4.NS.A (maximum number of blocks per direction is 4).

3) Set powers of the three spatial dependencies for the X-, Y-, Z-component in lines 143 to 145 of 4.NS.A. Their sum per component must not exceed 6.

4) Set selection switches for the selected terms in line 2 and the values of their coefficients in line 3 of 4.NS.A. If affine or power dependencies on the bifurcation parameter η of these coefficients are desired, set the appropriate constants CC and powers IX in lines 6 and 8 respectively.

5) Set the dimensions of ACIR, ACIT in BIF3A and FUN equal to the number of terms selected in Step 4). The lines which require this operation are marked by a star parade.

6) In subroutine FUN set up the final flow profiles in the section "setting of flow structure", again marked by a star parade. Make sure the coefficients CF, as selected in Step 4), are arranged in ascending order w.r.t. their indices.

The setup in the listings presented is for the study of flow IV (see Chapter 9).

The graphics and Fourier analysis programs are rather straightforward and technical in nature without any details of interest w.r.t. the problem under study. They are therefore not listed. The same goes for the transposing routine TRNSP (see Appendix B), since it is identical in its function and structure to the transposing section in the program ADV.

The following program chart (Figure 17) shows the sequence of execution of the programs with logical unit numbers for data transfer.

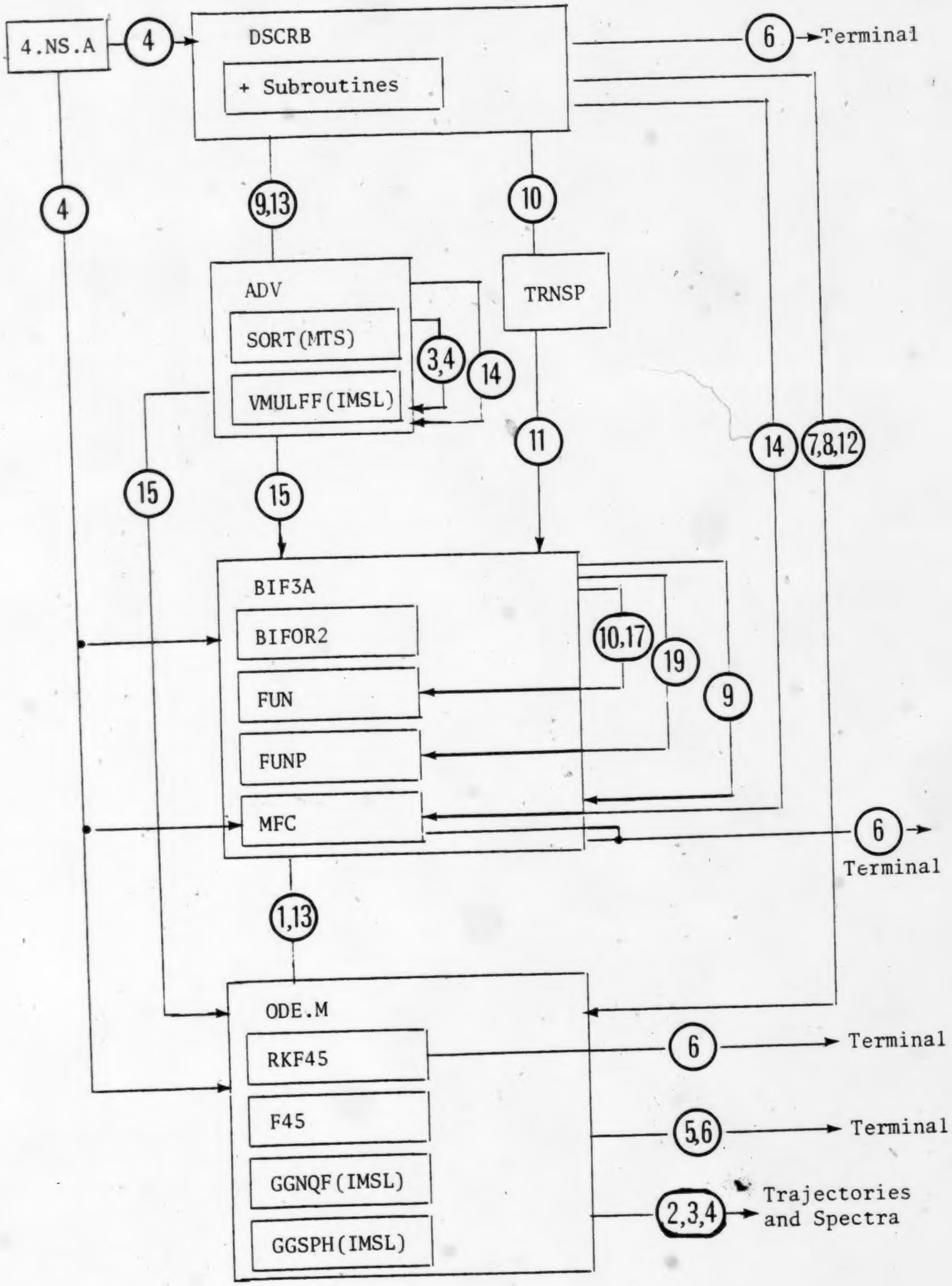

4.NS.A
4
DSCRB
+ Subroutines
6
Terminal
9,13
10
4
ADV
SORT(MTS)
VMULFF(IMSL)
3,4
14
TRNSP
11
14
7,8,12
15
15
BIF3A
BIFOR2
FUN
FUNP
MFC
10,17
19
9
6
Terminal
1,13
ODE.M
RKF45
F45
GGNQF(IMSL)
GGSPH(IMSL)
6
Terminal
5,6
Terminal
2,3,4
Trajectories and Spectra

FIGURE 17. Program Chart

Subroutine/Function Name	Line Number
SUBROUTINE SORT2	1 443.
FUNCTION MATINV	1 565.
SUBROUTINE QMTRXA	1 704.
SUBROUTINE QMTRXB	1 802.
SUBROUTINE QMTRXN	1 868.
FUNCTION QMLTI1	1 939.
FUNCTION QMLTI2	2 240.
SUBROUTINE FUN	2 875.
SUBROUTINE FUNP	3 088.
SUBROUTINE MFC	3 289.
SUBROUTINE F45	3 609.

Program Name	Line Number
DSCRB	1
ADV	2305
BIF3A	2422
ODE.M	3356

```
C PROGRAM DSCRB
C
C FUNCTION: DISCRETIZE A 3 DIMENSONAL DOMAIN IN FLUID
C           SPACE BY TRIANGULATION OF RECTANGULAR BLOCKS
C           ,CALCULATE THE ASS. FINITE ELEMENTS UNDER
C           THE CONDITION OF ZERO DIVERGENCE AND APPLY
C           THE GALERKIN METHOD TO GENERATE THE MATRIX
C           ELEMENTS OF A SYSTEM OF ODE'S WHICH IS THE
C           DISCRETIZED APPROXIMATION FOR A NONAUTONO-
C           MOUS NS-EQUATION FOR PERTURBATION VELOCITIES.
C
C SUBROUTINES USED: SORT2 ,MATINV ,QMTRXA ,QMTRXB ,QMTRXN
C                   ,QMLTI1 ,QMLTI2
C                   ,LINV3F ,VMULFF (BOTH IMSLLIB)
C
C LOG.UNITS ADDRESSED: 4 : INPUT  - PHYS. CONTROL PARAMETERS
C                      6 : OUTPUT - ERROR CODES ,CHECKS
C                      7 : OUTPUT - IC1
C                      8 : OUTPUT - IC2
C                      9 : OUTPUT - INVERTED ALHS
C                     10 : OUTPUT - LIN. RHS MATRICES MULTIPLIED
C                                   WITH INVERTED ALHS
C                     12 : OUTPUT - NONLIN. RHS MATRIX MULTIPLIED
C                                   WITH INVERTED ALHS
C                     13 : OUTPUT - NONLIN. RHS MATRIX WITH TRIPLE
C                                   INDICES (FROM ROUTINE QMTRXN)
C                     14 : OUTPUT - NODAL COORDINATES
C
C SYMBOLS USED: AMUL      = OUTPUT ARRAY FOR MULTIPLICATION OF
C                           LINEAR MATRICES WITH LHS INVERSE
C             ANLMUL      = SAME AS AMUL ,BUT FOR NONLIN.MATRIX
C             LC1 ,LC2    = DISTRIBUTION OF VELOCITY COMPONENTS
C                           IN NONLIN. FL.SPACE SUBDECOMPOSITION
C         IFX ,IFY ,IFZ = X ,Y ,Z - EXPONENTS FOR MEAN FLOW
C           XL ,YL ,ZL  = TEMP. STORAGE FOR VERTEX COORDS.
C            W ,D1 ,D2  = COEFFS. FOR FINITE ELEMENT POLYNOM
C                         IALS AND THEIR 1ST AND 2ND DERIVATIVES
C                  NL4  = TEMP. STORAGE FOR NODE COUNT
C             ANL ,TNL  = TEMP. STORAGES FOR NONLIN.MATRIX
C                         ELEMENTS WHEN CALCULATED IN QMTRXN
C               NCOUNT  = NODECOUNT
C               ICOUNT  = SWITCH FOR NODECOUNT
C               JCOUNT  = SWITCH FOR COLUMN COUNT IN NONLIN.
C                         MATRIX (BEFORE FL.SPACE SUBDECOMP.)
C                KCTCL  = COLUNM COUNT
C             C ,D ,E   = X ,Y ,Z - COORDS. FOR TETRAHEDRAL
C                         VERTICES
C          SX ,SY ,SZ   = SUMS OF THE X ,Y ,Z - COORDS.
C                         OVER THE FOUR TETRAHEDRAL VERTICES
C                  DET  = FUNCTIONAL DETERMINANT BETWEEN
```

```
        C                  BARYCENT. AND CARTES. COORDS.
        C     XNC ,YNC ,ZNC = X ,Y ,Z - COORDS. OF THE CENTROIDS
        C                  ON THE TETRAHEDRAL FACES ,I.E. THE
        C                  NODAL COORDS.
        C              ANLF = ARRAY FOR NONLIN.MATRIX AFTER FL.
        C                  SPACE SUBDECOMPOSITION
        C         IC1 ,IC2 = ARRAYS ASSIGNING THE INDICES IC1
        C                  ,IC2 OF THE APPROPRIATE VELOCITY
        C                  COMPONENTS IN THE PRODUCT VECTOR
        C                  TO THEIR COLUMN NOS. OF THE NONLIN.
        C                  MATRIX
        C                NC = ARRAY ASSIGNING THE COUNT NUMBER NC
        C                  OF A NODE TO ITS IDENTITY
        C              NCOL = ARRAY ASSIGNING THE COUNT NUMBER
        C                  OF COMBINATIONS OF A PAIR OF NODES
        C                  WITHIN A TETRAHEDRON TO THE PAIR
        C                  OF THEIR IDENTITY NUMBERS WITHIN
        C                  THIS TETRAHEDRON (ONLY USED FOR
        C                  COMBINATIONS ON MUTUAL INTERACTIONS)
        C              ALHS = MATRIX FOR LHS
        C     ACX ,ACY ,ACZ = MATRICES FOR ABSOLUTE X ,Y ,Z - COMP.
        C                  OF MEAN FLOW
        C              AVSC = MATRIX FOR VISCOSITY TERM
        C              AADV = ARRAY FOR MATRICES OF SPATIALLY DEPD.
        C                  COMP. OF MEAN FLOW
        C        LI ,LJ ,LK = LOG. VARIABLES CONTROLLING ALTER-
        C                  NATION BETWEEN TETRAHEDRAL CONFIG-
        C                  URATION A ,B PER BLOCK IN X ,Y ,Z-
        C                  DIRECTION
        C
  1           LOGICAL*1 LFMT(1) /'*'/
  2           IMPLICIT LOGICAL(L)
  3           INTEGER I, J, K, L1, L2, L3, LR, LC, LLC, LL2, LL3, T, NCOUNT,
             1        ICOUNT, LCC2, LCC3, LC23
        C
        C ------- STATEMENT FUNCTIONS ---------------------------
        C
        C  NODAL COORDINATES DETERMINED FROM VERTEX COMPONENTS A ,B :
        C
  4           CD(A,B) = (2*A + B) / 3
        C
        C  ROW AND COLUMN NOS. IN FLUID SPACE SUBDECOMPOSITION :
        C
  5           NCAR(I,J) = 3 * (I - 1) + J
  6           NCAC(I,J) = 6 * (I - 1) + J
  7           NNAC(I,J) = 9 * (I - 1) + J
        C
        C  NO. OF NODES FOR NOS. OF X ,Y ,Z - INCREMENTS M1 ,M2 ,M3 :
        C
  8           NCTMX(M1,M2,M3) = ((M1 - 1)*(M2 - 1)*(M3 - 1)*2 + (M1 - 2)*(M2 -
          11)*(M3 - 1) + (M1 - 1)*(M2 - 2)*(M3 - 1) + (M1 - 1)*(M2 - 1)*(M3 -
          2 2)) * 2
        C
        C------------------------------------------------------------
        C
        C***** ARRAYS FOR 3 RECT.BLOCKS ****************************
        C48   REAL ALHS(48,48)/2304*0./
        C48   REAL ACX (48,48)/2304*0./
        C48   REAL ACY (48,48)/2304*0./
        C48   REAL ACZ (48,48)/2304*0./
        C48   REAL AVSC(48,48)/2304*0./
        C48   REAL ANLF(276,48)/13248*0./
        C48   REAL AADV(48,48,3,3)/20736*0./
        C48   REAL AMUL(48,48)
        C48   REAL ANLMUL(276,48)
        C
        C TECH. PARAMETERS FOR LINV3F ,VMULFF WITH IC1 ,IC2 ,NC
        C
        C48   REAL BINV(1),WKAREA(96),D1INV/-1./
        C48   INTEGER NINV/48/,IAINV/48/,IJOBNV/1/,ICMUL/48/
        C48   INTEGER LMUL/48/,MMUL/48/,NMUL/48/,IAMUL/48/,IBMUL/48/
        C48   INTEGER LNL/276/,IANL/276/,ICNL/276/,IC1(276),IC2(276)
        C48   INTEGER NC(4,5,2,2,2)
        C***** ARRAYS FOR 2 RECT.BLOCKS ****************************
        C30   REAL ALHS(30,30)/900*0./
        C30   REAL ACX (30,30)/900*0./
        C30   REAL ACY (30,30)/900*0./
        C30   REAL ACZ (30,30)/900*0./
        C30   REAL AVSC(30,30)/900*0./
        C30   REAL ANLF(204,30)/6120*0./
        C30   REAL AADV(30,30,3,3)/8100*0./
        C30   REAL AMUL(30,30)
        C30   REAL ANLMUL(204,30)
        C
        C TECH. PARAMETERS FOR LINV3F ,VMULFF WITH IC1 ,IC2 ,NC
        C
        C30   REAL BINV(1),WKAREA(60),D1INV/-1./
        C30   INTEGER NINV/30/,IAINV/30/,IJOBNV/1/,ICMUL/30/
        C30   INTEGER LMUL/30/,MMUL/30/,NMUL/30/,IAMUL/30/,IBMUL/30/
        C30   INTEGER LNL/204/,IANL/204/,ICNL/204/,IC1(204),IC2(204)
        C30   INTEGER NC(4,5,3,2,2)
        C***** ARRAYS FOR 1 RECT.BLOCK  ****************************
    9         REAL ALHS(12,12) /144*0./
   10         REAL ACX(12,12) /144*0./
   11         REAL ACY(12,12) /144*0./
   12         REAL ACZ(12,12) /144*0./
   13         REAL AVSC(12,12) /144*0./
   14         REAL ANLF(78,12) /936*0./
   15         REAL AADV(12,12,3,3) /1296*0./
   16         REAL AMUL(12,12)
```

```
      REAL ANLMUL(78,12)
C
C TECH. PARAMETERS FOR LINV3F ,VMULFF WITH IC1 ,IC2 ,NC
C
      REAL BINV(1), WKAREA(24), D1INV /-1./
      INTEGER NINV /12/, IAINV /12/, IJOBNV /1/, ICMUL /12/
      INTEGER LMUL /12/, MMUL /12/, NMUL /12/, IAMUL /12/, IBMUL /12/
      INTEGER LNL /78/, IANL /78/, ICNL /78/, IC1(78), IC2(78)
      INTEGER NC(4,5,2,2,2)
C*********************************************************
      INTEGER IERINV, IERCX, IERCY, IERCZ, IERADV, IERVSC, IERNLF
C
      REAL SX(5), SY(5), SZ(5)
      REAL DET(5)
      DIMENSION NTL(4)
      DIMENSION XIN(5), YIN(5), ZIN(5)
      DIMENSION C(5,4), D(5,4), E(5,4)
      DIMENSION XNC(5,4), YNC(5,4), ZNC(5,4)
      DIMENSION NCOL(4,4)
      INTEGER LC1(6), LC2(6)
      DATA LC1 /1, 2, 3, 1, 1, 2/
      DATA LC2 /1, 2, 3, 2, 3, 3/
      COMMON /IFXYZ/ IFX(3), IFY(3), IFZ(3)
      COMMON /XYZ/ XL(4), YL(4), ZL(4)
      COMMON /WD12/ W(10,4), D1(4,3,4), D2(3,4)
      COMMON /MNL/ QT(7,7,7), NL(4)
      COMMON /ANLS/ ANL(3,6,4,4), TNL(3,4,4)
C
C USER - SUPPLIED PARAMETERS FOR DISCRETIZATION AND MEAN FLOW
C ARE READ FROM CONTROL PARAMETER FILE (LOG.UNIT 4) AND ECHOED
C
      READ (4'136000,LFMT) IMAX, JMAX, KMAX
      READ (4'138000,LFMT) (XIN(I),I=1,IMAX)
      READ (4'139000,LFMT) (YIN(J),J=1,JMAX)
      READ (4'140000,LFMT) (ZIN(K),K=1,KMAX)
      READ (4'143000,LFMT) IFX
      READ (4'144000,LFMT) IFY
      READ (4'145000,LFMT) IFZ
      WRITE (6,20) (XIN(I),I=1,IMAX)
      WRITE (6,20) (YIN(J),J=1,JMAX)
      WRITE (6,20) (ZIN(K),K=1,KMAX)
      WRITE (6,10) IFX
      WRITE (6,10) IFY
      WRITE (6,10) IFZ
   10 FORMAT (3(3X,I1))
   20 FORMAT (5(3X,F7.3))
C
C DIMENSION OF ODE SYSTEM IS CALCULATED
C
      NCTCL = NCTMX(IMAX,JMAX,KMAX)
```

```
 55          NCTC3 = NCTCL * 3
       C
       C
       C LOOPS K ,I ,J FOR INCREMENTS IN Z ,Y ,X - DIRECTION
       C WITH INITIALIZATIONS
       C
 56          LK = .FALSE.
 57          NCOUNT = 0
 58          KCTCL = 0
       C
 59          KMAXM1 = KMAX - 1
 60          DO 490 K = 1, KMAXM1
 61            LJ = LK
 62            Z = ZIN(K)
 63            ZZ = ZIN(K + 1)
       C
 64            JMAXM1 = JMAX - 1
 65            DO 480 J = 1, JMAXM1
 66              LI = LJ
 67              Y = YIN(J)
 68              YY = YIN(J + 1)
       C
 69              IMAXM1 = IMAX - 1
 70              DO 470 I = 1, IMAXM1
 71                X = XIN(I)
 72                XX = XIN(I + 1)
       C-------------------------------------------------------------
       C TETRAHEDRAL VERTEX COORDS. ARE COPIED INTO ARRAYS
       C C (X) ,D (Y) ,E (Z).
       C COORD.SUMS SX ,SY ,SZ AND DETERMINANTS DET ARE CALCULATED
       C-------------------------------------------------------------
       C SWITCH FOR SKIP TO OPERATIONS FOR CONFIGURATON B :
       C
 73                IF (LI) GO TO 30
       C
       C--- CONFIGURATION A (BLOCK CORNERS 1 ,2 ,4 ,7 SPLIT) -----
       C
       C      tetrahedron no 1a (0124)
       C
 74                C(1,1) = X
 75                C(1,2) = X
 76                C(1,3) = X
 77                C(1,4) = XX
       C
 78                SX(1) = 3 * X + XX
       C
 79                D(1,1) = Y
 80                D(1,2) = Y
 81                D(1,3) = YY
 82                D(1,4) = Y
```

```
251.          C
252.    83              SY(1) = 3 * Y + YY
253.          C
254.    84              E(1,1) = Z
255.    85              E(1,2) = ZZ
256.    86              E(1,3) = Z
257.    87              E(1,4) = Z
258.          C
259.    88              SZ(1) = 3 * Z + ZZ
260.          C
261.    89              DET(1) = -(X - XX) * (YY - Y) * (ZZ - Z)
262.          C
263.          C  TETRAHEDRON NO 2A (1237)
264.          C
265.    90              C(2,1) = X
266.    91              C(2,2) = X
267.    92              C(2,3) = X
268.    93              C(2,4) = XX
269.          C
270.    94              SX(2) = 3 * X + XX
271.          C
272.    95              D(2,1) = Y
273.    96              D(2,2) = YY
274.    97              D(2,3) = YY
275.    98              D(2,4) = YY
276.          C
277.    99              SY(2) = 3 * YY + Y
278.          C
279.   100              E(2,1) = ZZ
280.   101              E(2,2) = Z
281.   102              E(2,3) = ZZ
282.   103              E(2,4) = ZZ
283.          C
284.   104              SZ(2) = 3 * ZZ + Z
285.          C
286.   105              DET(2) = (X - XX) * (Y - YY) * (Z - ZZ)
287.          C
288.          C TETRAHEDRON NO 3A (1457)
289.          C
290.   106              C(3,1) = X
291.   107              C(3,2) = XX
292.   108              C(3,3) = XX
293.   109              C(3,4) = XX
294.          C
295.   110              SX(3) = 3 * XX + X
296.          C
297.   111              D(3,1) = Y
298.   112              D(3,2) = Y
299.   113              D(3,3) = Y
300.   114              D(3,4) = YY
```

```
        C
 115                SY(3) = 3 * Y + YY
        C
 116                E(3,1) = ZZ
 117                E(3,2) = Z
 118                E(3,3) = ZZ
 119                E(3,4) = ZZ
        C
 120                SZ(3) = 3 * ZZ + Z
        C
 121                DET(3) = -(X - XX) * (Y - YY) * (Z - ZZ)
        C
        C TETRAHEDRON NO 4A (2467)
        C
 122                C(4,1) = X
 123                C(4,2) = XX
 124                C(4,3) = XX
 125                C(4,4) = XX
        C
 126                SX(4) = 3 * XX + X
        C
 127                D(4,1) = YY
 128                D(4,2) = Y
 129                D(4,3) = YY
 130                D(4,4) = YY
        C
 131                SY(4) = 3 * YY + Y
        C
 132                E(4,1) = Z
 133                E(4,2) = Z
 134                E(4,3) = Z
 135                E(4,4) = ZZ
        C
 136                SZ(4) = 3 * Z + ZZ
        C
 137                DET(4) = (X - XX) * (Y - YY) * (Z - ZZ)
        C
        C TETRAHEDRON NO 5A (1247 INT)
        C
 138                C(5,1) = X
 139                C(5,2) = X
 140                C(5,3) = XX
 141                C(5,4) = XX
        C
 142                SX(5) = 2 * (XX + X)
        C
 143                D(5,1) = Y
 144                D(5,2) = YY
 145                D(5,3) = Y
 146                D(5,4) = YY
```

```
351.          C
352.    147             SY(5) = 2 * (Y + YY)
353.          C
354.    148             E(5,1) = ZZ
355.    149             E(5,2) = Z
356.    150             E(5,3) = Z
357.    151             E(5,4) = ZZ
358.          C
359.    152             SZ(5) = 2 * (ZZ + Z)
360.          C
361.    153             DET(5) = -2 * (X - XX) * (Y - YY) * (Z - ZZ)
362.          C
363.          C COORDS. OF NODES IN CONF.A ARE CALCULATED
364.          C
365.          C TETRAHEDRON NO 1A (0124)
366.          C
367.    154             XNC(1,1) = CD(X,XX)
368.    155             XNC(1,2) = CD(X,XX)
369.    156             XNC(1,3) = CD(X,XX)
370.    157             XNC(1,4) = X
371.          C
372.    158             YNC(1,1) = CD(Y,YY)
373.    159             YNC(1,2) = CD(Y,YY)
374.    160             YNC(1,3) = Y
375.    161             YNC(1,4) = CD(Y,YY)
376.          C
377.    162             ZNC(1,1) = CD(Z,ZZ)
378.    163             ZNC(1,2) = Z
379.    164             ZNC(1,3) = CD(Z,ZZ)
380.    165             ZNC(1,4) = CD(Z,ZZ)
381.          C
382.          C TETRAHEDRON NO 2A (1237)
383.          C
384.    166             XNC(2,1) = CD(X,XX)
385.    167             XNC(2,2) = CD(X,XX)
386.    168             XNC(2,3) = CD(X,XX)
387.    169             XNC(2,4) = X
388.          C
389.    170             YNC(2,1) = YY
390.    171             YNC(2,2) = CD(YY,Y)
391.    172             YNC(2,3) = CD(YY,Y)
392.    173             YNC(2,4) = CD(YY,Y)
393.          C
394.    174             ZNC(2,1) = CD(ZZ,Z)
395.    175             ZNC(2,2) = ZZ
396.    176             ZNC(2,3) = CD(ZZ,Z)
397.    177             ZNC(2,4) = CD(ZZ,Z)
398.          C
399.          C TETRAHEDRON NO 3A (1457)
400.          C
```

```
 178              XNC(3,1) = XX
 179              XNC(3,2) = CD(XX,X)
 180              XNC(3,3) = CD(XX,X)
 181              XNC(3,4) = CD(XX,X)
        C
 182              YNC(3,1) = CD(Y,YY)
 183              YNC(3,2) = CD(Y,YY)
 184              YNC(3,3) = CD(Y,YY)
 185              YNC(3,4) = Y
        C
 186              ZNC(3,1) = CD(ZZ,Z)
 187              ZNC(3,2) = ZZ
 188              ZNC(3,3) = CD(ZZ,Z)
 189              ZNC(3,4) = CD(ZZ,Z)
        C
        C TETRAHEDRON NO 4A (2467)
        C
 190              XNC(4,1) = XX
 191              XNC(4,2) = CD(XX,X)
 192              XNC(4,3) = CD(XX,X)
 193              XNC(4,4) = CD(XX,X)
        C
 194              YNC(4,1) = CD(YY,Y)
 195              YNC(4,2) = YY
 196              YNC(4,3) = CD(YY,Y)
 197              YNC(4,4) = CD(YY,Y)
        C
 198              ZNC(4,1) = CD(Z,ZZ)
 199              ZNC(4,2) = CD(Z,ZZ)
 200              ZNC(4,3) = CD(Z,ZZ)
 201              ZNC(4,4) = Z
        C
        C SKIP OVER OPERATIONS FOR CONF .B :
        C
 202              GO TO 40
        C
        C
        C--- CONFIGURATION B (BLOCK CORNERS 0 ,3 ,5 ,6 SPLIT) ----
        C
        C TETRAHEDRON NO 1B (0135)
        C
 203       30     C(1,1) = X
 204              C(1,2) = X
 205              C(1,3) = X
 206              C(1,4) = XX
        C
 207              SX(1) = 3 * X + XX
        C
 208              D(1,1) = Y
 209              D(1,2) = Y
```

```
210                D(1,3) = YY
211                D(1,4) = Y
          C
212                SY(1) = 3 * Y + YY
          C
213                E(1,1) = Z
214                E(1,2) = ZZ
215                E(1,3) = ZZ
216                E(1,4) = ZZ
          C
217                SZ(1) = 3 * ZZ + Z
          C
218                DET(1) = (X - XX) * (YY - Y) * (Z - ZZ)
          C
          C TETRAHEDRON NO 2B (0236)
          C
219                C(2,1) = X
220                C(2,2) = X
221                C(2,3) = X
222                C(2,4) = XX
          C
223                SX(2) = 3 * X + XX
          C
224                D(2,1) = Y
225                D(2,2) = YY
226                D(2,3) = YY
227                D(2,4) = YY
          C
228                SY(2) = 3 * YY + Y
          C
229                E(2,1) = Z
230                E(2,2) = Z
231                E(2,3) = ZZ
232                E(2,4) = Z
          C
233                SZ(2) = 3 * Z + ZZ
          C
234                DET(2) = -(X - XX) * (Y - YY) * (ZZ - Z)
          C
          C TETRAHEDRON NO 3B (0456)
          C
235                C(3,1) = X
236                C(3,2) = XX
237                C(3,3) = XX
238                C(3,4) = XX
          C
239                SX(3) = 3 * XX + X
          C
240                D(3,1) = Y
241                D(3,2) = Y
```

```
 242            D(3,3) = Y
 243            D(3,4) = YY
      C
 244            SY(3) = 3 * Y + YY
      C
 245            E(3,1) = Z
 246            E(3,2) = Z
 247            E(3,3) = ZZ
 248            E(3,4) = Z
      C
 249            SZ(3) = 3 * Z + ZZ
      C
 250            DET(3) = (X - XX) * (Y - YY) * (ZZ - Z)
      C
      C TETRAHEDRON NO 4B (3567)
      C
 251            C(4,1) = X
 252            C(4,2) = XX
 253            C(4,3) = XX
 254            C(4,4) = XX
      C
 255            SX(4) = 3 * XX + X
      C
 256            D(4,1) = YY
 257            D(4,2) = Y
 258            D(4,3) = YY
 259            D(4,4) = YY
      C
 260            SY(4) = 3 * YY + Y
      C
 261            E(4,1) = ZZ
 262            E(4,2) = ZZ
 263            E(4,3) = Z
 264            E(4,4) = ZZ
      C
 265            SZ(4) = 3 * ZZ + Z
      C
 266            DET(4) = (X - XX) * (Y - YY) * (Z - ZZ)
      C
      C TETRAHEDRON NO 5B (0356 INT)
      C
 267            C(5,1) = X
 268            C(5,2) = X
 269            C(5,3) = XX
 270            C(5,4) = XX
      C
 271            SX(5) = 2 * (X + XX)
      C
 272            D(5,1) = Y
 273            D(5,2) = YY
 274              D(5,3) = Y
 275              D(5,4) = YY
            C
 276              SY(5) = 2 * (Y + YY)
            C
 277              E(5,1) = Z
 278              E(5,2) = ZZ
 279              E(5,3) = ZZ
 280              E(5,4) = Z
            C
 281              SZ(5) = 2 * (Z + ZZ)
            C
 282              DET(5) = -2 * (X - XX) * (Y - YY) * (ZZ - Z)
            C
            C COORDINATES OF NODES IN CONFIGURATION B
            C
            C TETRAHEDRON NO 1B (O135)
            C
 283              XNC(1,1) = CD(X,XX)
 284              XNC(1,2) = CD(X,XX)
 285              XNC(1,3) = CD(X,XX)
 286              XNC(1,4) = X
            C
 287              YNC(1,1) = CD(Y,YY)
 288              YNC(1,2) = CD(Y,YY)
 289              YNC(1,3) = Y
 290              YNC(1,4) = CD(Y,YY)
            C
 291              ZNC(1,1) = ZZ
 292              ZNC(1,2) = CD(ZZ,Z)
 293              ZNC(1,3) = CD(ZZ,Z)
 294              ZNC(1,4) = CD(ZZ,Z)
            C
            C TETRAHEDRON NO 2B (O236)
            C
 295              XNC(2,1) = CD(X,XX)
 296              XNC(2,2) = CD(X,XX)
 297              XNC(2,3) = CD(X,XX)
 298              XNC(2,4) = X
            C
 299              YNC(2,1) = YY
 300              YNC(2,2) = CD(YY,Y)
 301              YNC(2,3) = CD(YY,Y)
 302              YNC(2,4) = CD(YY,Y)
            C
 303              ZNC(2,1) = CD(Z,ZZ)
 304              ZNC(2,2) = CD(Z,ZZ)
 305              ZNC(2,3) = Z
 306              ZNC(2,4) = CD(Z,ZZ)
            C
```

```
C TETRAHEDRON NO 3B (0456)
C
              XNC(3,1) = XX
              XNC(3,2) = CD(XX,X)
              XNC(3,3) = CD(XX,X)
              XNC(3,4) = CD(XX,X)
C
              YNC(3,1) = CD(Y,YY)
              YNC(3,2) = CD(Y,YY)
              YNC(3,3) = CD(Y,YY)
              YNC(3,4) = Y
C
              ZNC(3,1) = CD(Z,ZZ)
              ZNC(3,2) = CD(Z,ZZ)
              ZNC(3,3) = Z
              ZNC(3,4) = CD(Z,ZZ)
C
C TETRAHEDRON NO 4B (3567)
C
              XNC(4,1) = XX
              XNC(4,2) = CD(XX,X)
              XNC(4,3) = CD(XX,X)
              XNC(4,4) = CD(XX,X)
C
              YNC(4,1) = CD(YY,Y)
              YNC(4,2) = YY
              YNC(4,3) = CD(YY,Y)
              YNC(4,4) = CD(YY,Y)
C
              ZNC(4,1) = CD(ZZ,Z)
              ZNC(4,2) = CD(ZZ,Z)
              ZNC(4,3) = ZZ
              ZNC(4,4) = CD(ZZ,Z)
C
   40         CONTINUE
C-----------------------------------------------------------------
C
C T - LOOP ITERATES THROUGH THE FIVE TETRAHEDRONS PER BLOCK
C
              DO 460 T = 1, 5
C
C CORRECT VERTEX COORDS. ARE ASSIGNED TO THE CURRENT TETRAHEDRON
C BY COPYING C ,D ,E INTO THE TEMP. ARRAYS XL ,YL ,ZL
C
                DO 50 J1 = 1, 4
                  XL(J1) = C(T,J1)
                  YL(J1) = D(T,J1)
                  ZL(J1) = E(T,J1)
   50           CONTINUE
C
```

```
          C-----------------------------------------------------------
          C
          C ALL SUMMANDS FOR THE MATRIX ELEMENTS ARE CALCULATED BY
          C QMLTI1 AND STORED IN QT BY THEIR POWERS L1 ,L2 ,L3.
          C THE OFFSET OF ONE INDUCED BY K1 ,K2 ,K3 INCLUDES POWERS OF ZERO
          C
  338                   DO 60 K1 = 1, 7
  339                     DO 60 K2 = 1, 7
  340                       DO 60 K3 = 1, 7
  341                         L1 = K1 - 1
  342                         L2 = K2 - 1
  343                         L3 = K3 - 1
  344                         IF (L1 + L2 + L3 .GT. 6) GO TO 60
  345                         QT(K3,K2,K1) = QMLTI1(L3,L2,L1) * ABS(DET(T))
  346        60         CONTINUE
          C
          C-----------------------------------------------------------
          C
          C LIN. 10*10 SYSTEM FOR THE COEFFICIENTS OF THE FINITE ELEMENTS
          C IS SET UP BY SORT2 AND SOLVED BY MATINV (CALLED BY SORT2).
          C
  347                   CALL SORT2(SX(T), SY(T), SZ(T))
          C
          C-----------------------------------------------------------
          C
          C INITIALIZATION OF SWITCHES FOR NODECOUNT (ICOUNT) AND
          C NODE COMBINATION COUNT (JCOUNT).
          C
  348                   ICOUNT = 1
  349                   JCOUNT = 1
          C
          C------- FIRST TETRAHEDRAL NODE LOOP L1 : -------------------
          C
  350                   DO 450 L1 = 1, 4
          C
          C-----------------------------------------------------------
          C
          C STATEMENTS TO SKIP NODES IN THE SURFACE PLANES OF THE DOMAIN :
          C
          C SWITCH FOR PASSING TO SKIP STATEMENT SET FOR CONF. B :
          C
  351                      IF (LI) GO TO 70
          C
          C SKIP STATEMENT SET FOR CONF. A :
          C
  352                      IF (L1 .EQ. 4 .AND. T .EQ. 1 .AND. I .EQ. 1)
               1              GO TO 450
  353                      IF (L1 .EQ. 3 .AND. T .EQ. 1 .AND. J .EQ. 1)
               1              GO TO 450
  354                      IF (L1 .EQ. 2 .AND. T .EQ. 1 .AND. K .EQ. 1)
```

```
                  1           GO TO 450
         355               IF (L1 .EQ. 4 .AND. T .EQ. 2 .AND. I .EQ. 1)
                  1           GO TO 450
         356               IF (L1 .EQ. 4 .AND. T .EQ. 3 .AND. J .EQ. 1)
                  1           GO TO 450
         357               IF (L1 .EQ. 4 .AND. T .EQ. 4 .AND. K .EQ. 1)
                  1           GO TO 450
         358               IF (L1 .EQ. 1 .AND. T .EQ. 2 .AND. J .EQ. JMAXM1)
                  1           GO TO 450
         359               IF (L1 .EQ. 2 .AND. T .EQ. 2 .AND. K .EQ. KMAXM1)
                  1           GO TO 450
         360               IF (L1 .EQ. 1 .AND. T .EQ. 3 .AND. I .EQ. IMAXM1)
                  1           GO TO 450
         361               IF (L1 .EQ. 2 .AND. T .EQ. 3 .AND. K .EQ. KMAXM1)
                  1           GO TO 450
         362               IF (L1 .EQ. 1 .AND. T .EQ. 4 .AND. I .EQ. IMAXM1)
                  1           GO TO 450
         363               IF (L1 .EQ. 2 .AND. T .EQ. 4 .AND. J .EQ. JMAXM1)
                  1           GO TO 450
         364               GO TO 80
               C
               C SKIP STATEMENT SET FOR CONF. B :
               C
         365    70         IF (L1 .EQ. 3 .AND. T .EQ. 1 .AND. J .EQ. 1)
                  1           GO TO 450
         366               IF (L1 .EQ. 4 .AND. T .EQ. 1 .AND. I .EQ. 1)
                  1           GO TO 450
         367               IF (L1 .EQ. 3 .AND. T .EQ. 2 .AND. K .EQ. 1)
                  1           GO TO 450
         368               IF (L1 .EQ. 4 .AND. T .EQ. 2 .AND. I .EQ. 1)
                  1           GO TO 450
         369               IF (L1 .EQ. 3 .AND. T .EQ. 3 .AND. K .EQ. 1)
                  1           GO TO 450
         370               IF (L1 .EQ. 4 .AND. T .EQ. 3 .AND. J .EQ. 1)
                  1           GO TO 450
         371               IF (L1 .EQ. 1 .AND. T .EQ. 1 .AND. K .EQ. KMAXM1)
                  1           GO TO 450
         372               IF (L1 .EQ. 1 .AND. T .EQ. 2 .AND. J .EQ. JMAXM1)
                  1           GO TO 450
         373               IF (L1 .EQ. 1 .AND. T .EQ. 3 .AND. I .EQ. IMAXM1)
                  1           GO TO 450
         374               IF (L1 .EQ. 1 .AND. T .EQ. 4 .AND. I .EQ. IMAXM1)
                  1           GO TO 450
         375               IF (L1 .EQ. 2 .AND. T .EQ. 4 .AND. J .EQ. JMAXM1)
                  1           GO TO 450
         376               IF (L1 .EQ. 3 .AND. T .EQ. 4 .AND. K .EQ. KMAXM1)
                  1           GO TO 450
               C
               C------------------------------------------------------------
               C
```

```
       C TEMP. STORAGES FOR NONLIN. MATRIX ELEMENTS TNL .ANL ARE CLEARED
       C
377      80       DO 90 J3 = 1, 4
378                 DO 90 J2 = 1, 4
379                   DO 90 IR = 1, 3
380                     TNL(IR,J2,J3) = 0.
381                     DO 90 I6 = 1, 6
382      90       ANL(IR,I6,J2,J3) = 0.
       C
       C--------- SECOND TETRAHEDRAL NODE LOOP L2 : ----------------
       C
383               DO 340 L2 = 1, 4
       C
       C-------------------------------------------------------------
       C
       C STATEMENTS TO SKIP NODES IN THE SURFACE PLANES OF THE DOMAIN :
       C
       C SWITCH FOR PASSING TO SKIP STATEMENT SET FOR CONF. B :
       C
384                 IF (LI) GO TO 100
       C
       C SKIP STATEMENT SET FOR CONF. A :
       C
385                 IF (L2 .EQ. 4 .AND. T .EQ. 1 .AND. I .EQ. 1)
          1           GO TO 340
386                 IF (L2 .EQ. 3 .AND. T .EQ. 1 .AND. J .EQ. 1)
          1           GO TO 340
387                 IF (L2 .EQ. 2 .AND. T .EQ. 1 .AND. K .EQ. 1)
          1           GO TO 340
388                 IF (L2 .EQ. 4 .AND. T .EQ. 2 .AND. I .EQ. 1)
          1           GO TO 340
389                 IF (L2 .EQ. 4 .AND. T .EQ. 3 .AND. J .EQ. 1)
          1           GO TO 340
390                 IF (L2 .EQ. 4 .AND. T .EQ. 4 .AND. K .EQ. 1)
          1           GO TO 340
391                 IF (L2 .EQ. 1 .AND. T .EQ. 2 .AND. J .EQ. JMAXM1)
          1           GO TO 340
392                 IF (L2 .EQ. 2 .AND. T .EQ. 2 .AND. K .EQ. KMAXM1)
          1           GO TO 340
393                 IF (L2 .EQ. 1 .AND. T .EQ. 3 .AND. I .EQ. IMAXM1)
          1           GO TO 340
394                 IF (L2 .EQ. 2 .AND. T .EQ. 3 .AND. K .EQ. KMAXM1)
          1           GO TO 340
395                 IF (L2 .EQ. 1 .AND. T .EQ. 4 .AND. I .EQ. IMAXM1)
          1           GO TO 340
396                 IF (L2 .EQ. 2 .AND. T .EQ. 4 .AND. J .EQ. JMAXM1)
          1           GO TO 340
397                 GO TO 110
       C
       C SKIP STATEMENT SET FOR CONF. B :
```

```
      C
398    100          IF (L2 .EQ. 3 .AND. T .EQ. 1 .AND. J .EQ. 1)
         1             GO TO 340
399                 IF (L2 .EQ. 4 .AND. T .EQ. 1 .AND. I .EQ. 1)
         1             GO TO 340
400                 IF (L2 .EQ. 3 .AND. T .EQ. 2 .AND. K .EQ. 1)
         1             GO TO 340
401                 IF (L2 .EQ. 4 .AND. T .EQ. 2 .AND. I .EQ. 1)
         1             GO TO 340
402                 IF (L2 .EQ. 3 .AND. T .EQ. 3 .AND. K .EQ. 1)
         1             GO TO 340
403                 IF (L2 .EQ. 4 .AND. T .EQ. 3 .AND. J .EQ. 1)
         1             GO TO 340
404                 IF (L2 .EQ. 1 .AND. T .EQ. 1 .AND. K .EQ. KMAXM1)
         1             GO TO 340
405                 IF (L2 .EQ. 1 .AND. T .EQ. 2 .AND. J .EQ. JMAXM1)
         1             GO TO 340
406                 IF (L2 .EQ. 1 .AND. T .EQ. 3 .AND. I .EQ. IMAXM1)
         1             GO TO 340
407                 IF (L2 .EQ. 1 .AND. T .EQ. 4 .AND. I .EQ. IMAXM1)
         1             GO TO 340
408                 IF (L2 .EQ. 2 .AND. T .EQ. 4 .AND. J .EQ. JMAXM1)
         1             GO TO 340
409                 IF (L2 .EQ. 3 .AND. T .EQ. 4 .AND. K .EQ. KMAXM1)
         1             GO TO 340
      C
      C------------------------------------------------------------
      C
      C-------- THIRD TETRAHEDRAL NODE LOOP L3 : ------------------
      C
410    110          DO 330 L3 = 1, 4
      C
      C-----------------------------------------------------------
      C
      C STATEMENTS TO SKIP NODES IN THE SURFACE PLANES OF THE DOMAIN :
      C
      C SWITCH FOR PASSING TO SKIP STATEMENT SET FOR CONF. B :
      C
411                    IF (LI) GO TO 120
      C
      C SKIP STATEMENT SET FOR CONF. A :
      C
412                    IF (L3 .EQ. 4 .AND. T .EQ. 1 .AND. I .EQ. 1)
         1                GO TO 320
413                    IF (L3 .EQ. 3 .AND. T .EQ. 1 .AND. J .EQ. 1)
         1                GO TO 320
414                    IF (L3 .EQ. 2 .AND. T .EQ. 1 .AND. K .EQ. 1)
         1                GO TO 320
415                    IF (L3 .EQ. 4 .AND. T .EQ. 2 .AND. I .EQ. 1)
         1                GO TO 320
```

```
416                  IF (L3 .EQ. 4 .AND. T .EQ. 3 .AND. J .EQ. 1)
           1            GO TO 320
417                  IF (L3 .EQ. 4 .AND. T .EQ. 4 .AND. K .EQ. 1)
           1            GO TO 320
418                  IF (L3 .EQ. 1 .AND. T .EQ. 2 .AND. J .EQ. JMAXM1)
           1            GO TO 320
419                  IF (L3 .EQ. 2 .AND. T .EQ. 2 .AND. K .EQ. KMAXM1)
           1            GO TO 320
420                  IF (L3 .EQ. 1 .AND. T .EQ. 3 .AND. I .EQ. IMAXM1)
           1            GO TO 320
421                  IF (L3 .EQ. 2 .AND. T .EQ. 3 .AND. K .EQ. KMAXM1)
           1            GO TO 320
422                  IF (L3 .EQ. 1 .AND. T .EQ. 4 .AND. I .EQ. IMAXM1)
           1            GO TO 320
423                  IF (L3 .EQ. 2 .AND. T .EQ. 4 .AND. J .EQ. JMAXM1)
           1            GO TO 320
424                  GO TO 130
      C
      C SKIP STATEMENT SET FOR CONF. B :
      C
425     120          IF (L3 .EQ. 3 .AND. T .EQ. 1 .AND. J .EQ. 1)
           1            GO TO 320
426                  IF (L3 .EQ. 4 .AND. T .EQ. 1 .AND. I .EQ. 1)
           1            GO TO 320
427                  IF (L3 .EQ. 3 .AND. T .EQ. 2 .AND. K .EQ. 1)
           1            GO TO 320
428                  IF (L3 .EQ. 4 .AND. T .EQ. 2 .AND. I .EQ. 1)
           1            GO TO 320
429                  IF (L3 .EQ. 3 .AND. T .EQ. 3 .AND. K .EQ. 1)
           1            GO TO 320
430                  IF (L3 .EQ. 4 .AND. T .EQ. 3 .AND. J .EQ. 1)
           1            GO TO 320
431                  IF (L3 .EQ. 1 .AND. T .EQ. 1 .AND. K .EQ. KMAXM1)
           1            GO TO 320
432                  IF (L3 .EQ. 1 .AND. T .EQ. 2 .AND. J .EQ. JMAXM1)
           1            GO TO 320
433                  IF (L3 .EQ. 1 .AND. T .EQ. 3 .AND. I .EQ. IMAXM1)
           1            GO TO 320
434                  IF (L3 .EQ. 1 .AND. T .EQ. 4 .AND. I .EQ. IMAXM1)
           1            GO TO 320
435                  IF (L3 .EQ. 2 .AND. T .EQ. 4 .AND. J .EQ. JMAXM1)
           1            GO TO 320
436                  IF (L3 .EQ. 3 .AND. T .EQ. 4 .AND. K .EQ. KMAXM1)
           1            GO TO 320
      C
      C-------------------------------------------------------------
      C
      C COUNTSKIP FOR ITERATIONS PAST THE FIRST ONE WITHIN
      C THE THREE TETRAHEDRAL NODE LOOPS L1 ,L2 ,L3 :
      C
      130             IF (ICOUNT .EQ. 0) GO TO 300
C
C------------------------------------------------------------
C
C COUNTSKIP SETS FOR PREVIOUSLY COUNTED NODES :
C
C SWITCH FOR COUNSKIP SET OF CONF. B :
C
                      IF (LI) GO TO 140
C
C COUNTSKIP SET OF CONF. A :
C
                      IF (J .GE. 2 .AND. T .EQ. 1 .AND. L3 .EQ. 2)
     1                   GO TO 170
                      IF (J .GE. 2 .AND. T .EQ. 1 .AND. L3 .EQ. 3)
     1                   GO TO 180
                      IF (I .GE. 2 .AND. T .EQ. 1 .AND. L3 .EQ. 4)
     1                   GO TO 190
                      IF (I .GE. 2 .AND. T .EQ. 2 .AND. L3 .EQ. 4)
     1                   GO TO 200
                      IF (J .GE. 2 .AND. T .EQ. 3 .AND. L3 .EQ. 4)
     1                   GO TO 210
                      IF (K .GE. 2 .AND. T .EQ. 4 .AND. L3 .EQ. 4)
     1                   GO TO 220
                      GO TO 150
C
C COUNTSKIP SET OF CONF. B :
C
      140             IF (J .GE. 2 .AND. T .EQ. 1 .AND. L3 .EQ. 3)
     1                   GO TO 230
                      IF (I .GE. 2 .AND. T .EQ. 1 .AND. L3 .EQ. 4)
     1                   GO TO 240
                      IF (K .GE. 2 .AND. T .EQ. 2 .AND. L3 .EQ. 3)
     1                   GO TO 250
                      IF (I .GE. 2 .AND. T .EQ. 2 .AND. L3 .EQ. 4)
     1                   GO TO 260
                      IF (K .GE. 2 .AND. T .EQ. 3 .AND. L3 .EQ. 3)
     1                   GO TO 270
                      IF (J .GE. 2 .AND. T .EQ. 3 .AND. L3 .EQ. 4)
     1                   GO TO 280
C
C------------------------------------------------------------
C
C COUNTSKIP FOR INTERIOR (I.E. 5TH) TETRAHEDRON PER BLOCK :
C
      150             IF (T .EQ. 5) GO TO 290
C
C NODECOUNT AND AND ITS STORAGE IN NC FOR LATER IDENTIFICATION
C BY VERTEX (L3) ,TETRAHEDRON (T) AND X ,Y ,Z-INCREMENTS (I,J,K)
C AND IN NL BY VERTEX (L3)
```

```
C
                      NCOUNT = NCOUNT + 1
                      NC(L3,T,I,J,K) = NCOUNT
                      NL(L3) = NCOUNT
C
C-------------------------------------------------------------
C
C COUNT NUMBERS ON THE INTERIOR NODES PER BLOCK ARE TEMPO-
C RARILY STORED IN NTL TO BE ASSIGNED LATER TO THE 5TH TET.
C
C SWITCH FOR CONF. B :
C
                      IF (LI) GO TO 160
C
C STATEMENT SET FOR CONF. A :
C
                      IF (T .EQ. 4 .AND. L3 .EQ. 3) NTL(1) = NCOUNT
                      IF (T .EQ. 3 .AND. L3 .EQ. 3) NTL(2) = NCOUNT
                      IF (T .EQ. 2 .AND. L3 .EQ. 3) NTL(3) = NCOUNT
                      IF (T .EQ. 1 .AND. L3 .EQ. 1) NTL(4) = NCOUNT
                      GO TO 300
C
C STATEMENT SET FOR CONF. B :
C
  160                 IF (T .EQ. 4 .AND. L3 .EQ. 4) NTL(1) = NCOUNT
                      IF (T .EQ. 3 .AND. L3 .EQ. 2) NTL(2) = NCOUNT
                      IF (T .EQ. 2 .AND. L3 .EQ. 2) NTL(3) = NCOUNT
                      IF (T .EQ. 1 .AND. L3 .EQ. 2) NTL(4) = NCOUNT
                      GO TO 300
C
C-------------------------------------------------------------
C
C NODAL COORDS. AND THEIR COUNTNO. ARE WRITTEN INTO LOG.UNIT 14
C
                      WRITE (14) XNC(T,L3), YNC(T,L3), ZNC(T,L3), NL(L3)
C
C-------------------------------------------------------------
C
C ENDPOINTS OF COUNTSKIPS FOR PREV. COUNTED NODES.
C THE NODES ARE REASSIGNED THEIR PREV. COUNTNO. BY USING NC
C AND COPYING THEM INTO NL.
C
C ENDPOINT SET FOR CONF. A :
C
  170                 NL(L3) = NC(1,1,I,J,K - 1)
                      GO TO 300
  180                 NL(L3) = NC(1,2,I,J - 1,K)
                      GO TO 300
  190                 NL(L3) = NC(1,3,I - 1,J,K)
                      GO TO 300
  200          NL(L3) = NC(1,4,I - 1,J,K)
               GO TO 300
  210          NL(L3) = NC(2,4,I,J - 1,K)
               GO TO 300
  220          NL(L3) = NC(3,4,I,J,K - 1)
               GO TO 300
C
C ENDPOINT SET FOR CONF. B :
C
  230          NL(L3) = NC(1,2,I,J - 1,K)
               GO TO 300
  240          NL(L3) = NC(1,3,I - 1,J,K)
               GO TO 300
  250          NL(L3) = NC(2,2,I,J,K - 1)
               GO TO 300
  260          NL(L3) = NC(1,4,I - 1,J,K)
               GO TO 300
  270          NL(L3) = NC(2,3,I,J,K - 1)
               GO TO 300
  280          NL(L3) = NC(2,4,I,J - 1,K)
               GO TO 300
C
C--------------------------------------------------------------
C
C ENDPOINT FOR COUNTSKIP ON 5TH TETRAHEDRON.
C COUNTNOS. ARE ASSIGNED TO THE 5TH TETRAHEDRON BY COPYING
C THEM FROM NTL TO NL
C
  290          NL(L3) = NTL(L3)
C
  300          CONTINUE
C
C--------------------------------------------------------------
C
C QMTRXN CALCULATES NONLIN. MATRIX ELEMENTS FOR EACH TRIPLE
C COMBINATION L1,L2,L3 OF NODES IN THE INTERIOR OF THE DOMAIN
C
               CALL QMTRXN(L1, L2, L3)
C
C--------------------------------------------------------------
C
C COUNT AND STORAGE OF NODAL PAIR COMBINATIONS PER TETRAHEDRON.
C COUNTSKIP FOR ITERATIONS PAST ONE IN THE TETRATEDRAL LOOPS
C AND DOUBLE COUNT OF NODAL COMBINATIONS.
C
               IF (JCOUNT .EQ. 0 .OR. NL(L2) .GE. NL(L3))
     1            GO TO 310
               KCTCL = KCTCL + 1
               NCOL(L3,L2) = KCTCL
  310          CONTINUE
```

```
        C
        C-----------------------------------------------------------------
        C
        C LINEAR MATRIX ELEMENTS ARE CALCULATED BY QMTRXA ,QMTRXB
        C ACCORDING TO NODAL COMBINATION L1,L2 :
        C
  499     320             IF (L3 .EQ. 4) CALL QMTRXA(NCTC3, L1, L2, AADV)
  500                     IF (L3 .EQ. 4) CALL QMTRXB(NCTC3, L1, L2, ALHS, ACX,
             1                 ACY, ACZ, AVSC)
        C
        C-----------------------------------------------------------------
        C
        C---------- NODAL LOOP L3 ENDS : ---------------------------------
        C
  501     330             CONTINUE
        C
        C--------- NODAL COUNT IS SWITCHED OFF : ----------------------
        C
  502                     ICOUNT = 0
        C
        C-----------------------------------------------------------------
        C
        C---------- NODAL LOOP L2 ENDS : ---------------------------------
        C
  503     340             CONTINUE
        C
        C--------- NODAL COMBINATION COUNT IS SWITCHED OFF : -----------
        C
  504                     JCOUNT = 0
        C
        C-----------------------------------------------------------------
        C
        C COPY ANL INTO LEFT PART OF ANLF WITH PROPER ROW (NNR) AND
        C COLUMN (NNC) NOS.
        C
  505                     NLL1 = NL(L1)
        C
        C LC-LOOP ITERATES THROUGH COLUMNS OF 3*6 FL.SPACE SUBDECOM
        C POSITIONS STORED IN ANL :
        C
  506                     DO 350 LC = 1, 6
        C
        C CALCULATE CORRECT COLUMN NO. NNC FOR ANLF :
        C
  507                     NNC = NCAC(NLL1,LC)
        C
        C CALCULATE INDEX ARRAYS IC1 ,IC2 AS FUNCTIONS OF NNC :
        C
  508                     IC1(NNC) = NCAR(NLL1,LC1(LC))
  509                     IC2(NNC) = NCAR(NLL1,LC2(LC))
```

```
          C
          C LR-LOOP ITERATES THROUGH ROWS OF 3*6 FL.SPACE SUBDECOM
          C POSITIONS STORED IN ANL :
          C
  510                DO 350 LR = 1, 3
          C
          C SELECT CORRECT ROW (LR) AND COLUMN (LC) INDICES WITHIN ANL :
          C
  511                  IF (LR + LC .EQ. 7) GO TO 350
  512                  IF (LR .NE. LC .AND. LC .LE. 3) GO TO 350
          C
          C CALCULATE CORRECT ROW NO. NNR FOR ANLF :
          C
  513                  NNR = NCAR(NLL1,LR)
          C
          C COPY ELEMENT FROM ANL INTO ANLF :
          C
  514                  AFN = ANL(LR,LC,L1,L1)
  515                  ANLF(NNC,NNR) = AFN + ANLF(NNC,NNR)
  516      350       CONTINUE
          C
          C-----------------------------------------------------------
          C
          C NODAL LOOP L2 IS REPEATED AS LL2 :
          C
  517                DO 440 LL2 = 1, 4
          C
          C-----------------------------------------------------------
          C
          C STATEMENTS TO SKIP NODES IN THE SURFACE PLANES OF THE DOMAIN :
          C
          C SWITCH FOR PASSING TO SKIP STATEMENT SET FOR CONF. B :
          C
  518                 IF (LI) GO TO 360
          C
          C SKIP STATEMENT SET FOR CONF. A :
          C
  519                 IF (LL2 .EQ. 4 .AND. T .EQ. 1 .AND. I .EQ. 1)
               1         GO TO 440
  520                 IF (LL2 .EQ. 3 .AND. T .EQ. 1 .AND. J .EQ. 1)
               1         GO TO 440
  521                 IF (LL2 .EQ. 2 .AND. T .EQ. 1 .AND. K .EQ. 1)
               1         GO TO 440
  522                 IF (LL2 .EQ. 4 .AND. T .EQ. 2 .AND. I .EQ. 1)
               1         GO TO 440
  523                 IF (LL2 .EQ. 4 .AND. T .EQ. 3 .AND. J .EQ. 1)
               1         GO TO 440
  524                 IF (LL2 .EQ. 4 .AND. T .EQ. 4 .AND. K .EQ. 1)
               1         GO TO 440
  525                 IF (LL2 .EQ. 1 .AND. T .EQ. 2 .AND. J .EQ. JMAXM1)
```

```
1 151.          1              GO TO 440
1 152.   526                IF (LL2 .EQ. 2 .AND. T .EQ. 2 .AND. K .EQ. KMAXM1)
1 153.          1              GO TO 440
1 154.   527                IF (LL2 .EQ. 1 .AND. T .EQ. 3 .AND. I .EQ. IMAXM1)
1 155.          1              GO TO 440
1 156. . 528                IF (LL2 .EQ. 2 .AND. T .EQ. 3 .AND. K .EQ. KMAXM1)
1 157.          1              GO TO 440
1 158.   529                IF (LL2 .EQ. 1 .AND. T .EQ. 4 .AND. I .EQ. IMAXM1)
1 159.          1              GO TO 440
1 160.   530                IF (LL2 .EQ. 2 .AND. T .EQ. 4 .AND. J .EQ. JMAXM1)
1 161.          1              GO TO 440
1 162.   531                GO TO 370
1 163.        C
1 164.        C SKIP STATEMENT SET FOR CONF. B :
1 165.        C
1 166.   532    360         IF (LL2 .EQ. 3 .AND. T .EQ. 1 .AND. J .EQ. 1)
1 167.          1              GO TO 440
1 168.   533                IF (LL2 .EQ. 4 .AND. T .EQ. 1 .AND. I .EQ. 1)
1 169.          1              GO TO 440
1 170.   534                IF (LL2 .EQ. 3 .AND. T .EQ. 2 .AND. K .EQ. 1)
1 171.          1              GO TO 440
1 172.   535                IF (LL2 .EQ. 4 .AND. T .EQ. 2 .AND. I .EQ. 1)
1 173.          1              GO TO 440
1 174.   536                IF (LL2 .EQ. 3 .AND. T .EQ. 3 .AND. K .EQ. 1)
1 175.          1              GO TO 440
1 176.   537                IF (LL2 .EQ. 4 .AND. T .EQ. 3 .AND. J .EQ. 1)
1 177.          1              GO TO 440
1 178.   538                IF (LL2 .EQ. 1 .AND. T .EQ. 1 .AND. K .EQ. KMAXM1)
1 179.          1              GO TO 440
1 180.   539                IF (LL2 .EQ. 1 .AND. T .EQ. 2 .AND. J .EQ. JMAXM1)
1 181.          1              GO TO 440
1 182.   540                IF (LL2 .EQ. 1 .AND. T .EQ. 3 .AND. I .EQ. IMAXM1)
1 183.          1              GO TO 440
1 184.   541                IF (LL2 .EQ. 1 .AND. T .EQ. 4 .AND. I .EQ. IMAXM1)
1 185.          1              GO TO 440
1 186.   542                IF (LL2 .EQ. 2 .AND. T .EQ. 4 .AND. J .EQ. JMAXM1)
1 187.          1              GO TO 440
1 188.   543                IF (LL2 .EQ. 3 .AND. T .EQ. 4 .AND. K .EQ. KMAXM1)
1 189.          1              GO TO 440
1 190.        C
1 191.        C-------------------------------------------------------------
1 192.        C
1 193.        C NODAL LOOP L3 IS REPEATED AS LL3 :
1 194.        C
1 195.   544    370         DO 430 LL3 = 1, 4
1 196.        C
1 197.        C-------------------------------------------------------------
1 198.        C
1 199.        C STATEMENTS TO SKIP NODES IN THE SURFACE PLANES OF THE DOMAIN :
1 200.        C
```

```
          C SWITCH FOR PASSING TO SKIP STATEMENT SET FOR CONF. B :
          C
   545                        IF (LI) GO TO 380
          C
          C SKIP STATEMENT SET FOR CONF. A :
          C
   546                        IF (LL3 .EQ. 4 .AND. T .EQ. 1 .AND. I .EQ. 1)
               1                 GO TO 430
   547                        IF (LL3 .EQ. 3 .AND. T .EQ. 1 .AND. J .EQ. 1)
               1                 GO TO 430
   548                        IF (LL3 .EQ. 2 .AND. T .EQ. 1 .AND. K .EQ. 1)
               1                 GO TO 430
   549                        IF (LL3 .EQ. 4 .AND. T .EQ. 2 .AND. I .EQ. 1)
               1                 GO TO 430
   550                        IF (LL3 .EQ. 4 .AND. T .EQ. 3 .AND. J .EQ. 1)
               1                 GO TO 430
   551                        IF (LL3 .EQ. 4 .AND. T .EQ. 4 .AND. K .EQ. 1)
               1                 GO TO 430
   552                        IF (LL3 .EQ. 1 .AND. T .EQ. 2 .AND. J .EQ. JMAXM1)
               1                 GO TO 430
   553                        IF (LL3 .EQ. 2 .AND. T .EQ. 2 .AND. K .EQ. KMAXM1)
               1                 GO TO 430
   554                        IF (LL3 .EQ. 1 .AND. T .EQ. 3 .AND. I .EQ. IMAXM1)
               1                 GO TO 430
   555                        IF (LL3 .EQ. 2 .AND. T .EQ. 3 .AND. K .EQ. KMAXM1)
               1                 GO TO 430
   556                        IF (LL3 .EQ. 1 .AND. T .EQ. 4 .AND. I .EQ. IMAXM1)
               1                 GO TO 430
   557                        IF (LL3 .EQ. 2 .AND. T .EQ. 4 .AND. J .EQ. JMAXM1)
               1                 GO TO 430
   558                        GO TO 390
          C
          C SKIP STATEMENT SET FOR CONF. B :
          C
   559      380               IF (LL3 .EQ. 3 .AND. T .EQ. 1 .AND. J .EQ. 1)
               1                 GO TO 430
   560                        IF (LL3 .EQ. 4 .AND. T .EQ. 1 .AND. I .EQ. 1)
               1                 GO TO 430
   561                        IF (LL3 .EQ. 3 .AND. T .EQ. 2 .AND. K .EQ. 1)
               1                 GO TO 430
   562                        IF (LL3 .EQ. 4 .AND. T .EQ. 2 .AND. I .EQ. 1)
               1                 GO TO 430
   563                        IF (LL3 .EQ. 3 .AND. T .EQ. 3 .AND. K .EQ. 1)
               1                 GO TO 430
   564                        IF (LL3 .EQ. 4 .AND. T .EQ. 3 .AND. J .EQ. 1)
               1                 GO TO 430
   565                        IF (LL3 .EQ. 1 .AND. T .EQ. 1 .AND. K .EQ. KMAXM1)
               1                 GO TO 430
   566                        IF (LL3 .EQ. 1 .AND. T .EQ. 2 .AND. J .EQ. JMAXM1)
               1                 GO TO 430
```

```
 567                    IF (LL3 .EQ. 1 .AND. T .EQ. 3 .AND. I .EQ. IMAXM1)
            1              GO TO 430
 568                    IF (LL3 .EQ. 1 .AND. T .EQ. 4 .AND. I .EQ. IMAXM1)
            1              GO TO 430
 569                    IF (LL3 .EQ. 2 .AND. T .EQ. 4 .AND. J .EQ. JMAXM1)
            1              GO TO 430
 570                    IF (LL3 .EQ. 3 .AND. T .EQ. 4 .AND. K .EQ. KMAXM1)
            1              GO TO 430
          C
          C-----------------------------------------------------------------
          C
          C NODAL PAIR COMBINATIONS (I.E. MUTUAL INTERACTIONS) ARE
          C REDETERMINED. THEIR COUNTNOS. ARE PICKED FROM NCOL AND
          C USED TO CALCULATE COLUMN NOS. (NCC) IN THE RIGHT PART OF
          C ANLF (BEYOND COL.NO. = NCTCL*6). INDEX ARRAYS IC1 ,IC2
          C ARE CALCULATED AS A FUNCTION OF NCC AND ELEMENTS OF TNL
          C ARE COPIED INTO THE RIGHT PART OF ANLF ACCORDING TO THEIR
          C PROPER ROW (NRR) AND COLUMN (NCC) NOS.
          C
          C SKIP IDENTICAL (I.E. SELFINTERACTIONS) AND REPEATING NODAL
          C PAIR COMBINATIONS :
          C
 571        390         IF (NL(LL2) .GE. NL(LL3)) GO TO 430
          C
          C INITIALIZE COLUMN COUNT (LC23) FOR 3*9 FL.SPACE SUBDECOMPO
          C SITIONS :
          C
 572                    LC23 = 0
          C
          C LOOPS LCC2 AND LCC3 ITERATE THROUGH THE THREE COMPONENTS
          C IN THE FL.SPACE SUBDECOMPOSITION OF THE TWO FACTORS IN
          C THE PRODUCT VECTOR OF UNKNOWN VELOCITIES (TO BE MULTIPLIED
          C WITH THE NONLIN. MATRIX) :
          C
 573                    DO 420 LCC2 = 1, 3
 574                      DO 410 LCC3 = 1, 3
          C
          C INCREMENT COLUMN COUNT LC23 :
          C
 575                          LC23 = LC23 + 1
          C
          C CALCULATE COLUMN NO. NCC FOR NONLIN.MATRIX ANLF AND ASS.
          C WITH PROPER ENTRIES OF INDEX ARRAYS IC1 ,IC2 FOR PRODUCT
          C VECTOR OF UNKNOWN VELOCITIES :
          C
 576                          NCC = NNAC(NCOL(LL3,LL2),LC23) + NCTCL * 6
 577                          IC1(NCC) = NCAR(NL(LL2),LCC2)
 578                          IC2(NCC) = NCAR(NL(LL3),LCC3)
          C
          C LR-LOOP ITERATES THROUGH ROWS OF 3*9 FL.SPACE SUBDECOMPO
```

```
        C SITIONS :
        C
 579                         DO 400 LR = 1, 3
        C
        C SELECT CORRECT ROW NO. WITHIN 3*9 SUBDECOMPOSITION :
        C
 580                           IF (LR .NE. LCC2) GO TO 400
        C
        C CALCULATE ROW NO. NRR FOR NONLIN. MATRIX ANLF :
        C
 581                           NRR = NCAR(NLL1,LR)
        C
        C SELECT PROPER ELEMENT FROM TNL ACCORDING TO LCC3 :
        C
 582                           AFN = TNL(LCC3,LL3,LL2)
        C
        C IF THE COMPONENTS LCC2 AND LCC3 OF THE VELOCITY SUBDECOMPO
        C SITIONS ARE IDENTICAL ,ADD FROM TNL THE ELEMENT SYMMETRIC IN
        C LL2 ,LL3 (IN THIS CASE ,VELOCITIES IN PRODUCT VECTOR COMMUTE) :
        C
 583                           IF (LCC2 .EQ. LCC3) AFN = AFN + TNL(LCC3,LL2,
           1                   LL3)
        C
        C COPY ELEMENT INTO NONLIN. MATRIX ANLF :
        C
 584                           ANLF(NCC,NRR) = AFN + ANLF(NCC,NRR)
 585      400                CONTINUE
 586      410              CONTINUE
 587      420            CONTINUE
 588      430          CONTINUE
 589      440        CONTINUE
        C
        C-------------------------------------------------------------
        C
        C-------- NODAL LOOP L1 ENDS : -------------------------------
        C
 590      450        CONTINUE
        C
        C--------- END OF TETRAHEDRAL LOOP T : -----------------------
        C
 591      460      CONTINUE
        C
        C-------------------------------------------------------------
        C
        C END OF LOOPS I ,J ,K FOR X ,Y ,Z - INCREMENTATION
        C WITH LOG. SWITCHES LI ,LJ ,LK FOR ALTERNATION OF CONF. A ,B
        C
 592                LI = .NOT. LI
 593      470     CONTINUE
 594              LJ = .NOT. LJ
 480   CONTINUE
       LK = .NOT. LK
 490 CONTINUE
C
C-----------------------------------------------------------------
C
C COLUMN NO. OF NONLIN. MATRIX AND NODE NOS. OF THE LAST
C TETRAHEDRON ARE PRINTED OUT FOR CHECK.
C INDEX ARRAYS IC1 ,IC2 ARE WRITTEN ON LOG.UNITS 7 ,8.
C
      WRITE (6,500) NCC, (NL(I),I=1,4)
 500 FORMAT ('MAX.COL-# ', I4, 3X, 'MAX.NODE-#=', 4(2X,I3))
C
      WRITE (7) IC1
      WRITE (8) IC2
C
C-----------------------------------------------------------------
C
C MATRIX ALHS ON LHS OF SYSTEM IS INVERTED BY LINV3F.
C OUTPUT IS WRITTEN INTO LOG.UNIT 9
C
      CALL LINV3F(ALHS, BINV, IJOBNV, NINV, IAINV, D1INV, D2INV, WKAREA,
     1     IERINV)
      WRITE (6,590) IERINV
      IF (IERINV .NE. O) STOP 10
      DO 510 IR = 1, NCTC3
 510 WRITE (9) (ALHS(IC,IR),IC=1,NCTC3)
C
C-----------------------------------------------------------------
C
C INVERTED LHS-MATRIX (IN ARRAY ALHS NOW) IS MULTIPLIED WITH
C ALL MATRICES ON THE RHS OF THE SYSTEM BY VMULFF.
C OUTPUT IS WRITTEN INTO LOG.UNITS 10 (LIN.) AND 12 (NONLIN.)
C
      CALL VMULFF(ACX, ALHS, LMUL, MMUL, NMUL, IAMUL, IBMUL, AMUL,
     1     ICMUL, IERCX)
      WRITE (6,600) IERCX
      IF (IERCX .NE. O) STOP 11
      DO 520 IR = 1, NCTC3
 520 WRITE (10) (AMUL(IC,IR),IC=1,NCTC3)
C
      CALL VMULFF(ACY, ALHS, LMUL, MMUL, NMUL, IAMUL, IBMUL, AMUL,
     1     ICMUL, IERCY)
      WRITE (6,610) IERCY
      IF (IERCY .NE. O) STOP 22
      DO 530 IR = 1, NCTC3
 530 WRITE (10) (AMUL(IC,IR),IC=1,NCTC3)
C
      CALL VMULFF(ACZ, ALHS, LMUL, MMUL, NMUL, IAMUL, IBMUL, AMUL,
     1     ICMUL, IERCZ)
 1 401.      618          WRITE (6,620) IERCZ
 1 402.      619          IF (IERCZ .NE. O) STOP 33
 1 403.      620          DO 540 IR = 1, NCTC3
 1 404.      621      540 WRITE (10) (AMUL(IC,IR),IC=1,NCTC3)
 1 405.             C
 1 406.      622          CALL VMULFF(AVSC, ALHS, LMUL, MMUL, NMUL, IAMUL, IBMUL, AMUL,
 1 407.                  1     ICMUL, IERVSC)
 1 408.      623          WRITE (6,630) IERVSC
 1 409.      624          IF (IERVSC .NE. O) STOP 44
 1 410.      625          DO 550 IR = 1, NCTC3
 1 411.      626      550 WRITE (10) (AMUL(IC,IR),IC=1,NCTC3)
 1 412.             C
 1 413.      627          DO 570 JMUL = 1, [illegible]
 1 414.      628            DO 570 IMUL = 1, 3
 1 415.      629              CALL VMULFF(AADV(1,1,IMUL,JMUL), ALHS, LMUL, MMUL, NMUL,
 1 416.                  1          IAMUL, IBMUL, AMUL, ICMUL, IERADV)
 1 417.      630              WRITE (6,650) IERADV, IMUL, JMUL
 1 418.      631              IF (IERADV .NE. O) STOP 99
 1 419.      632              DO 560 IR = 1, NCTC3
 1 420.      633      560     WRITE (10) (AMUL(IC,IR),IC=1,NCTC3)
 1 421.      634      570 CONTINUE
 1 422.             C
 1 423.      635          CALL VMULFF(ANLF, ALHS, LNL, MMUL, NMUL, IANL, IBMUL, ANLMUL,
 1 424.                  1     ICNL, IERNLF)
 1 425.      636          WRITE (6,640) IERNLF
 1 426.      637          IF (IERNLF .NE. O) STOP 55
 1 427.      638          DO 580 IR = 1, NCTC3
 1 428.      639      580 WRITE (12) (ANLMUL(IC,IR),IC=1,NCC)
 1 429.             C
 1 430.      640      590 FORMAT ('IERINV=', 2X, I3)
 1 431.      641      600 FORMAT ('IERCX =', 2X, I3)
 1 432.      642      610 FORMAT ('IERCY =', 2X, I3)
 1 433.      643      620 FORMAT ('IERCZ =', 2X, I3)
 1 434.      644      630 FORMAT ('IERVSC=', 2X, I3)
 1 435.      645      640 FORMAT ('IERNLF=', 2X, I3)
 1 436.      646      650 FORMAT ('IERADV=', 2X, I3, 2X, I1, 1X, I1)
 1 437.             C
 1 438.      647          STOP
 1 439.             CDS   DEBUG SUBCHK
 1 440.      648          END
```

```
C___________________________________________________________________
C
      SUBROUTINE SORT2(SX, SY, SZ)
C
C FUNCTION: ASSEMBLE AND SOLVE THE LIN. 10*10 SYSTEMS FOR
C           THE COEFFICIENTS OF THE FINITE ELEMENTS ASS.
C           WITH EACH ONE OF THE FOUR NODES OF A TETRAHEDRON
C           AND DETERMINE THE COEFFS. OF THEIR FIRST AND
C           SECOND PARTIAL DERIVATIVES.
C
C SUBROUTINES USED: MATINV (PERFORMS MATRIX INVERSION)
C
C SYMBOLS USED: A    = MATRIX OF KNOWN RHS VALUES
C               B    = COPY OF A - USED FOR TECH. TRANSFERS
C               W    = LHS VECTOR - BEFORE INVERSION OF THE SYSTEM
C                    = COEFFICIENTS - AFTER INVERSION
C               D1   = COEFFICIENTS OF FIRST DERIVATIVE
C               D2   = COEFFICIENTS OF SECOND DERIVATIVE
C         XL ,YL ,ZL = TETRAHEDRAL VERTEX COORDS.
C         SX ,SY ,SZ = VERTEX COORD.SUMS
C
      REAL A(10,10), B(10,10)
      COMMON /XYZ/ XL(4), YL(4), ZL(4)
      COMMON /WD12/ W(10,4), D1(4,3,4), D2(3,4)
C
C ARRAY A IS CLEARED FROM PREV. TETRAHEDRON :
C
      DO 10 IC = 1, 10
        DO 10 IR = 1, 10
   10 A(IR,IC) = 0.
C
C ELEMENTS OF A WHICH REMAIN THE SAME FOR ALL FOUR NODES
C ARE DETERMINED :
C
      DO 20 I1 = 1, 4
        A(I1,1) = 1.
        A(I1,2) = (SX - XL(I1)) / 3
        A(I1,3) = (SY - YL(I1)) / 3
        A(I1,4) = (SZ - ZL(I1)) / 3
        A(I1,5) = (SX - XL(I1)) * (SX - XL(I1)) / 9
        A(I1,6) = (SY - YL(I1)) * (SY - YL(I1)) / 9
        A(I1,7) = (SZ - ZL(I1)) * (SZ - ZL(I1)) / 9
        A(I1,8) = (SX - XL(I1)) * (SY - YL(I1)) / 9
        A(I1,9) = (SY - YL(I1)) * (SZ - ZL(I1)) / 9
        A(I1,10) = (SZ - ZL(I1)) * (SX - XL(I1)) / 9
   20 CONTINUE
C
      A(5,2) = 1.
      A(6,3) = 1.
      A(7,4) = 1.
      A(8,2) = 1. / 6
      A(9,3) = 1. / 6
      A(10,4) = 1. / 6
      A(8,5) = SX / 12
      A(9,6) = SY / 12
      A(10,7) = SZ / 12
      A(8,8) = SY / 24
      A(9,9) = SZ / 24
      A(10,10) = SX / 24
      A(9,8) = SX / 24
      A(10,9) = SY / 24
      A(8,10) = SZ / 24
C
C ELEMENTS OF A AND W VARYING BETWEEN THE FOUR NODES
C ARE ASSEMBLED - L ITERATES THROUGH ALL FOUR NODES :
C
      DO 50 L = 1, 4
C
        DO 30 I2 = 1, 10
   30   W(I2,L) = 0.
        W(L,L) = 1.
C
        A(5,5) = 2 * (SX - XL(L)) / 3
        A(6,6) = 2 * (SY - YL(L)) / 3
        A(7,7) = 2 * (SZ - ZL(L)) / 3
C
        A(5,8) = (SY - YL(L)) / 3
        A(6,8) = (SX - XL(L)) / 3
        A(6,9) = (SZ - ZL(L)) / 3
        A(7,9) = (SY - YL(L)) / 3
        A(5,10) = (SZ - ZL(L)) / 3
        A(7,10) = (SX - XL(L)) / 3
C
C COPY A INTO B :
C
        DO 40 LC = 1, 10
          DO 40 LR = 1, 10
   40   B(LR,LC) = A(LR,LC)
C
C MATINV INVERTS B ,MULTIPLIES WITH W AND STORES
C OUTPUT VECTOR IN W (OPERATONS FOR NODE NO. L) :
C
        IERR = MATINV(B,W(1,L),10,10)
        IF (IERR .NE. 0) STOP 77
C
C COEFFS. OF THE DERIVATIVES ARE DETERMINED :
C
        D1(1,1,L) = W(2,L)
        D1(1,2,L) = W(3,L)
        D1(1,3,L) = W(4,L)
```

```
 1 541.           C
 1 542.      56          D1(2,1,L) = 2 * W(5,L)
 1 543.      57          D1(2,2,L) = W(8,L)
 1 544.      58          D1(2,3,L) = W(10,L)
 1 545.           C
 1 546.      59          D1(3,1,L) = W(8,L)
 1 547.      60          D1(3,2,L) = 2 * W(6,L)
 1 548.      61          D1(3,3,L) = W(5,L)
 1 549.           C
 1 550.      62          D1(4,1,L) = W(10,L)
 1 551.      63          D1(4,2,L) = W(9,L)
 1 552.      64          D1(4,3,L) = 2 * W(7,L)
 1 553.           C
 1 554.      65          D2(1,L) = 2 * W(5,L)
 1 555.      66          D2(2,L) = 2 * W(6,L)
 1 556.      67          D2(3,L) = 2 * W(7,L)
 1 557.           C
 1 558.      68        50 CONTINUE
 1 559.           C
 1 560.      69          RETURN
 1 561.           CDS    DEBUG SUBCHK
 1 562.      70          END
```

```
        C_______________________________________________________________
        C
   1          FUNCTION MATINV(DA, DB, M, N)
        C
        C     Purpose: solution of N simultaneous equations in N unknowns
        C       given by the matrix equation  A * x = B
        C       where  A is a N by N matrix and X & B are N-element vectors
        C       The solution (vector x) is returned in vector B (over-
        C       writting the input) and the inverse of matrix A is returned
        C       in A, again overwritting the input matrix.
        C
        C     Function Arguments:
        C       DA     = input M by M real array containing LHS of equation
        C                  of which only N rows by N columns are used
        C              = returned N by N values contain the inverse of the
        C                 input array -- array A**(-1)
        C       DB     = input N real vector containing RHS of equation
        C              = returned values contain the solution (array x)
        C       M      = number of rows that matrix DA is DIMENSIONED
        C                  Must be <= 100.  Must be given correctly other-
        C                  wise the result will be incorrect.!!
        C       N      = number of equations (or unknowns). Must be <= M
        C
        C     Function returns status of solution
        C        0   =  all ok
        C        1   =  error in M or N, not in range 1-100
        C       -1   =  solution is singular
        C
        C     Method: uses Gaussian pivoting on rows to move the largest
        C       element onto the diagonal element
        C
        C     Routine can be converted to Double Precision by un-commenting
        C      the  CCs in cols 1 & 2 (& commenting in the following line
        C      or lines if necessary)
        C
        C       -------------------------------------------------
        C
        C
        CC    IMPLICIT REAL*8 (D)
   2          IMPLICIT LOGICAL*4(L)
        C
        C       Run-time dimensioning for 'dummy' arguments
        C        the array DA and vector DB.
        C
   3          DIMENSION DA(M,M), DB(N)
        C
   4          DIMENSION INDEX(100,2), L(100)
        C
   5          EQUIVALENCE (DSWAP,DPIVOT), (IROW,JROW), (ICOL,JCOL)
        C
        C       -----------------------------------------------------
```

```
     C
     C      Check if M and N is valid, in range 1-100. !!
     C
  6        MATINV = 1
  7        IF (M .LE. 0 .OR. M .GT. 100 .OR. N .LE. 0 .OR. N .GT. 100)
          1   GO TO 120
  8        MATINV = -1
     C
     C      Initialization
     C
  9        DO 10 I = 1, N
 10          L(I) = .FALSE.
 11     10 CONTINUE
     C
     CC    DMAX = 1.1D-60
 12        DMAX = 1.1E-60
     C
 13        DO 90 I = 1, N
     C       Search for the largest pivot element.
     CC      DMAX = 1.0D-15 * DMAX
     CC      IF( DMAX .EQ. 0.0D0 ) DMAX = 1.1D-60
 14          DMAX = 1.0E-7 * DMAX
 15          IF (DMAX .EQ. 0.0E0) DMAX = 1.1E-60
 16          LSING = .TRUE.
 17          DO 30 II = 1, N
 18            IF (L(II)) GO TO 30
 19            DO 20 JJ = 1, N
 20              IF (L(JJ)) GO TO 20
     CC          DX = DABS( DA(II,JJ) )
 21              DX = ABS(DA(II,JJ))
 22              IF (DMAX .GT. DX) GO TO 20
 23              IROW = II
 24              ICOL = JJ
 25              DMAX = DX
 26              LSING = .FALSE.
 27     20     CONTINUE
 28     30   CONTINUE
     C       Is matrix singular ??  If so, MATINV is -1.
 29          IF (LSING) GO TO 120
 30          L(ICOL) = .TRUE.
 31          INDEX(I,1) = IROW
 32          INDEX(I,2) = ICOL
 33          IF (IROW .EQ. ICOL) GO TO 50
     C       Interchange rows to put pivot on diagonal
 34          DO 40 JJ = 1, N
 35            DSWAP = DA(IROW,JJ)
 36            DA(IROW,JJ) = DA(ICOL,JJ)
 37            DA(ICOL,JJ) = DSWAP
 38     40   CONTINUE
 39          DSWAP = DB(IROW)
```

```
        DB(IROW) = DB(ICOL)
        DB(ICOL) = DSWAP
C         Divide pivot row by pivot element
   50   DPIVOT = DA(ICOL,ICOL)
        DA(ICOL,ICOL) = 1.0D0
        DO 60 JJ = 1, N
          DA(ICOL,JJ) = DA(ICOL,JJ) / DPIVOT
   60   CONTINUE
        DB(ICOL) = DB(ICOL) / DPIVOT
C        Reduce non-pivot rows
        DO 80 II = 1, N
          IF (II .EQ. ICOL) GO TO 80
          DTEMP = DA(II,ICOL)
          DA(II,ICOL) = 0.0D0
          DO 70 JJ = 1, N
            DA(II,JJ) = DA(II,JJ) - DA(ICOL,JJ) * DTEMP
   70     CONTINUE
          DB(II) = DB(II) - DB(ICOL) * DTEMP
   80   CONTINUE
   90 CONTINUE
C
C       Interchange columns to get inverse matrix
C
      DO 110 J = 1, N
        IN = N + 1 - J
        IF (INDEX(IN,1) .EQ. INDEX(IN,2)) GO TO 110
        JROW = INDEX(IN,1)
        JCOL = INDEX(IN,2)
        DO 100 I = 1, N
          DSWAP = DA(I,JROW)
          DA(I,JROW) = DA(I,JCOL)
          DA(I,JCOL) = DSWAP
  100   CONTINUE
  110 CONTINUE
C
      MATINV = 0
C
  120 RETURN
      END
      C____________________________________________________________
      C
  1         SUBROUTINE QMTRXA(NCTC3, L1, L2, AADV)
      C
      C FUNCTION: ASSEMBLE ALL 9 MATRICES STEMMING FROM THE ADVECTIVE
      C           AND SPATIALLY DEPENDENT TERMS IN THE MEAN FLOW
      C           ,I.E. ALL LIN. RHS-MATRICES WITH DIAGONAL AND
      C           NONDIAGONAL TERMS AS WELL.
      C
      C SUBROUTINES USED: QMLTI2
      C
      C SYMBOLS USED: AADV    = OUTPUT ARRAY HOLDING ALL 9 ADVECTIVE
      C                         MATRICES
      C           W ,D1 ,D2 = COEFFS. OF THE POLYNOMIALS OF THE
      C                         FINITE ELEMENTS AND THEIR 1ST AND
      C                         2ND DERIVATIVES
      C              NL       = TEMP. STORAGE FOR NODE COUNTS
      C        IFX ,IFY ,IFZ = USER-SUPPLIED EXPONENTS OF (SPATIAL)
      C                         X ,Y ,Z-DEPENDENCIES OF THE MEAN FLOW
      C                     E = TECH. ARRAY HOLDING FOUR TIMES UNITY
      C
  2         REAL AADV(NCTC3,NCTC3,3,3)
  3         COMMON /WD12/ W(10,4), D1(4,3,4), D2(3,4)
  4         COMMON /MNL/ QT(7,7,7), NL(4)
  5         COMMON /IFXYZ/ IFX(3), IFY(3), IFZ(3)
  6         DIMENSION E(4)
  7         DATA E /4*1./
      C-------------------------------------------------------------
      C STATEMENT FUNCTION: CALCULATE ROW- AND COLUMN NOS.
      C
  8         NCAI(I,J) = 3 * (I - 1) + J
      C-------------------------------------------------------------
      C
      C I1-LOOP ITERATES THROUGH THE THREE COMPONENTS OF THE MEAN FLOW :
      C
  9         DO 40 I1 = 1, 3
      C
      C I2-LOOP ITERATES THROUGH THE THREE SPATIAL DEPENDENCIES OF THE
      C MEAN FLOW (NO TERMS WITH MIXED DEPENDENCIES ARE USED) :
      C
 10           DO 30 I2 = 1, 3
      C
      C EXPONENTS OF SPATIAL FLOW DEPENDENCIES ARE INITIALIZED :
      C
 11             IPX = 1
 12             IPY = 1
 13             IPZ = 1
      C
      C USER-SUPPLIED EXPONENTS ARE SUBSTITUTED :
      C
```

```
1 752.     14            IF (I2 .EQ. 1) IPX = IFX(I1)
1 753.     15            IF (I2 .EQ. 2) IPY = IFY(I1)
1 754.     16            IF (I2 .EQ. 3) IPZ = IFZ(I1)
1 755.          C
1 756.          C DIFFERENTIATED (TE1) AND PURELY DIAGONAL (TE2) ELEMENTS W.R.T.
1 757.          C THE 3*3 FL.SPACE SUBDECOMPOSITION ARE CALCULATED BY QMLTI2 :
1 758.          C
1 759.     17            TE1 = QMLTI2(1,1,10,10,E,W(1,L2),W(1,L1),IPX,IPY,IPZ) * IPX *
1 760.              1      IPY * IPZ
1 761.     18            TE2 = QMLTI2(I2 + 1,I2 + 1,4,10,E,D1(1,I1,L2),W(1,L1),IPX,IPY,
1 762.              1      IPZ)
1 763.          C
1 764.          C------------------------------------------------------------
1 765.          C DIAGONAL ELEMENTS W.R.T. THE 3*3 FL.SPACE SUBDECOMPOSITION ARE
1 766.          C COPIED INTO AADV BY ID-LOOP :
1 767.          C
1 768.     19            DO 20 ID = 1, 3
1 769.          C
1 770.          C CALCULATE ROW (NRD) AND COLUMN (NCD) NOS. FOR MATRIX IN AADV :
1 771.          C
1 772.     20              NRD = NCAI(NL(L1),ID)
1 773.     21              NCD = NCAI(NL(L2),ID)
1 774.          C
1 775.     22              IF (I1 .EQ. ID .AND. I2 .EQ. ID) GO TO 10
1 776.     23              AADV(NCD,NRD,I2,I1) = TE2 + AADV(NCD,NRD,I2,I1)
1 777.     24              GO TO 20
1 778.     25      10      TE12 = TE1 + TE2
1 779.     26              AADV(NCD,NRD,I2,I1) = TE12 + AADV(NCD,NRD,I2,I1)
1 780.     27      20    CONTINUE
1 781.          C--------------------------------------------------------------
1 782.          C NONDIAGONAL ELEMENTS OF 3*3 FL.SPACE SUBDECOMPOSITION ARE
1 783.          C COPIED INTO AADV :
1 784.          C
1 785.          C CALCULATE ROW (NR1) AND COLUMN (NC2) NOS. FOR MATRIX IN AADV :
1 786.          C
1 787.     28            NR1 = NCAI(NL(L1),I1)
1 788.     29            NC2 = NCAI(NL(L2),I2)
1 789.          C
1 790.     30            IF (I1 .NE. I2) AADV(NC2,NR1,I2,I1) = TE1 + AADV(NC2,NR1,I2,
1 791.              1      I1)
1 792.          C------------------------------------------------------------
1 793.          C
1 794.     31      30   CONTINUE
1 795.     32      40 CONTINUE
1 796.          C
1 797.     33         RETURN
1 798.          CDS     DEBUG SUBCHK
1 799.     34         END
```

```
          C_____________________________________________________________
          C
   1            SUBROUTINE QMTRXB(NCTC3, L1, L2, ALHS, ACX, ACY, ACZ, AVSC)
          C
          C FUNCTION: ASSEMBLE MATRICES FOR LHS (ALHS)
          C           ,ABSOLUTE ADVECTIVE MEAN FLOW TERMS (ACX ,ACY
          C           ,ACZ) AND VISCOSITY (AVSC).
          C
          C SUBROUTINES USED: QMLTI2
          C
          C SYMBOLS USED: ALHS   = MATRIX ON LHS
          C       ACX ,ACY ,ACZ = MATRICES FOR ABSOLUTE ADVECTIVE TERMS
          C                       OF X ,Y ,Z-COMPONENTS OF THE MEAN FLOW
          C                  TA = TEMP. ARRAY FOR ELEMENTS OF ACX
          C                       ,ACY ,ACZ
          C         W ,D1 ,D2 = COEFFS. OF THE POLYNOMIALS OF THE
          C                       FINITE ELEMENTS AND THEIR 1ST AND
          C                       2ND DERIVATIVES
          C                  NL = TEMP. STORAGE FOR NODE COUNTS
          C                   E = TECH. ARRAY HOLDING FOUR TIMES UNITY
          C
   2            REAL ALHS(NCTC3,NCTC3), ACX(NCTC3,NCTC3), ACY(NCTC3,NCTC3),
               1     ACZ(NCTC3,NCTC3), AVSC(NCTC3,NCTC3), TA(3), E(4)
   3            DATA E /4*1./
   4            COMMON /WD12/ W(10,4), D1(4,3,4), D2(3,4)
   5            COMMON /MNL/ QT(7,7,7), NL(4)
          C-------------------------------------------------------------
          C STATEMENT FUNCTION FOR CALCULATION OF ROW AND COLUMN NOS.
          C
   6            NCAI(I,J) = 3 * (I - 1) + J
          C-------------------------------------------------------------
          C CALCULATE ELEMENTS FOR VISCOSITY (TV) ,LHS (TL) AND ADVECTIVE
          C ABSOLUTE MEAN FLOW TERMS (TA) BY QMLTI2 :
          C
   7            TV = QMLTI2(1,1,1,10,E,D2(1,L2),W(1,L1),1,1,1) + QMLTI2(1,1,1,10,
               1E,D2(2,L2),W(1,L1),1,1,1) + QMLTI2(1,1,1,10,E,D2(3,L2),W(1,L1),1,
               21,1)
          C
   8            TL = QMLTI2(1,1,10,10,E,W(1,L2),W(1,L1),1,1,1)
          C
   9            DO 10 IL = 1, 3
  10         10 TA(IL) = QMLTI2(1,1,4,10,E,D1(1,IL,L2),W(1,L1),1,1,1)
          C
          C-----------------------------------------------------------
          C COPY THE ELEMENTS CALCULATED ABOVE INTO THEIR APPROPRIATE
          C ARRAYS :
          C
  11            DO 20 ID = 1, 3
          C
          C DETERMINE ROW (NCAR) AND COLUMN (NCAC) NOS. IN THESE ARRAYS :
```

```
          C
12              NCAR = NCAI(NL(L1),ID)
13              NCAC = NCAI(NL(L2),ID)
          C
14              ALHS(NCAC,NCAR) = TL + ALHS(NCAC,NCAR)
15              ACX(NCAC,NCAR) = TA(1) + ACX(NCAC,NCAR)
16              ACY(NCAC,NCAR) = TA(2) + ACY(NCAC,NCAR)
17              ACZ(NCAC,NCAR) = TA(3) + ACZ(NCAC,NCAR)
18              AVSC(NCAC,NCAR) = TV + AVSC(NCAC,NCAR)
          C
19         20 CONTINUE
          C---------------------------------------------------------------
          C
20            RETURN
          CDS   DEBUG SUBCHK
21            END
```

```
C_______________________________________________________________
C
      SUBROUTINE QMTRXN(L1, L2, L3)
C
C FUNCTION: DETERMINE THE THREE NONLINEAR MATRIX ELEMENTS
C           PER FL.SPACE SUBDECOMPOSITION ,WRITE THEM INTO
C           TEMP. ARRAY TNL (FOR ASSEMBLY OF 3*9 FL.SPACE
C           SUBDECOMPOSITION IN MAIN LINE) AND LOG. UNIT 13
C           (FOR LATER ASSEMBLY OF ADVECTIVE MATRICES OF
C           2ND AND/OR 3RD PERTURBATION) AND ASSEMBLE 3*6
C           FL.SPACE SUBDECOMPOSITION FOR SELFINTERACTIONS.
C
C SUBROUTINES USED: QMLTI2
C
C LOG.UNITS ADDRESSED: 13 : OUTPUT - NONLIN. RHS MATRIX WITH TRIPLE
C                                    INDICES
C
C SYMBOLS USED:      TN = TEMP. ARRAY FOR TECH. TRANSFER
C                    NL = TEMP. STORAGE FOR NODE COUNTS
C             W ,D1 ,D2 = COEFFS. OF THE POLYNOMIALS OF THE
C                         FINITE ELEMENTS AND THEIR 1ST AND
C                         2ND DERIVATIVES
C                   ANL = TEMP. ARRAY FOR TRANSFER OF THE
C                         FOUR 3*6 FL.SPACE SUBDECOMPOSIT
C                         IONS PER TETRAHEDRON INTO THE
C                         NONLIN. MATRIX IN MAIN LINE
C                   TNL = TEMP. ARRAY FOR TRANSFER OF THE
C                         THREE ELEMENTS FOR THE 3*9 FL.
C                         SPACE SUBDECOMPOSITION INTO THE
C                         NONLIN. MATRIX IN MAIN LINE
C
      REAL TN(3)
      COMMON /WD12/ W(10,4), D1(4,3,4), D2(3,4)
      COMMON /MNL/ QT(7,7,7), NL(4)
      COMMON /ANLS/ ANL(3,6,4,4), TNL(3,4,4)
C---------------------------------------------------------------
C STATEMENT FUNCTION: CALCULATE MATRIX INDEX NOS.
C
      NCAI(I,J) = 3 * (I - 1) + J
C---------------------------------------------------------------
C ALL THREE NONLIN. MATRIX ELEMENTS ARE CALCULATED BY QMLTI2
C AND TRANSFERRED INTO COLUMN 1 TO 3 OF THE 3*6 SUBDECOMPOSI
C TIONS IN ANL AND INTO TNL. INDEX NOS. NR ,N2 ,N3 ARE CALCUL
C ATED FOR THE THREE NODES ASS. WITH L1 ,L2 ,L3 AND WRITTEN
C INTO LOG. UNIT 13 WITH THEIR APPROPRIATE MATRIX ELEMENT.
C
      DO 10 IN = 1, 3
        TN(IN) = QMLTI2(1,4,10,10,D1(1,IN,L2),W(1,L1),W(1,L3),1,1,1)
        TNL(IN,L3,L2) = TN(IN)
        N3 = NCAI(NL(L3),IN)
        ANL(IN,IN,L3,L2) = TN(IN)
        DO 10 IRN = 1, 3
          NR = NCAI(NL(L1),IRN)
          N2 = NCAI(NL(L2),IRN)
          WRITE (13) TN(IN), N3, N2, NR
   10 CONTINUE
C----------------------------------------------------------------------
C MATRIX ELEMENTS ARE TRANSFERRED INTO COLUMNS 4 TO 6 IN THE
C 3*6 SUBDECOMPOSITIONS WITHIN ANL.
C
      DO 20 IC = 4, 6
        DO 20 IR = 1, 3
          IF (IC + IR .EQ. 7) GO TO 20
          IO = IC - IR - 1
          ANL(IR,IC,L3,L2) = TN(IO)
   20 CONTINUE
C----------------------------------------------------------------------
C
      RETURN
CDS   DEBUG SUBCHK
      END
```

```
C______________________________________________________________
C
      FUNCTION QMLTI1(M, N, O).
C
C FUNCTION: EVALUATE INTEGRALS OF (X**M)*(Y**N)*(Z**O) OVER
C           TETRAHEDRONS BY TRANSFORMING INTO BARYCENTRIC
C           COORDS. AND USING HAMMMER'S FORMULA.
C           THE POWER SUM M+N+O MUST NOT EXCEED 6.
C
C SYMBOLS USED: C     = TECH. ARRAY FOR COEFFS. ARISING FROM
C                       TRANSFORMATION BETWEEN CARTES. AND
C                       BARYCENT. COORDS.
C        XL ,YL ,ZL = X ,Y ,Z-COORDS. OF TETRAHEDRAL VERTICES
C             XNFAC = FACTORIALS FROM 0 TO 9 INDEXED WITH AN
C                       OFFSET OF 1
C
      REAL QMLTI1, SUM
      INTEGER M, N, O, N1, N2, N3, IN, IM, IO
      COMMON /XYZ/ XL(4), YL(4), ZL(4)
      DIMENSION C(6,4), XNFAC(10)
      DATA XNFAC /1.0, 1.0, 2.0, 6.0, 24.0, 120.0, 720., 5040., 40320.,
     1     362880./
C
      M1 = M + 1
      MN = M + N
      MN1 = M + N + 1
      MNO = M + N + O
C
C SKIP AND FLAG FOR POWER SUMS EQ. O OR GT. 6 :
C
      IF (MNO .EQ. O) GO TO 180
      IF (MNO .GT. 6) GO TO 190
C--------------------------------------------------------------
C INITIALIZE INDICES FOR XNFAC WITH AN OFFSET OF 1 :
C AND CARTES. VERTEX COORDS. (XL,YL,ZL) ARE DETERMINED AND
C STORED IN C :
C
      DO 60 L = 1, 3
        IF (M .EQ. O) GO TO 20
        DO 10 IM = 1, M
          C(IM,L) = XL(L) - XL(4)
   10   C(IM,4) = XL(4)
   20   IF (N .EQ. O) GO TO 40
        DO 30 IN = M1, MN
          C(IN,L) = YL(L) - YL(4)
   30   C(IN,4) = YL(4)
   40   IF (O .EQ. O) GO TO 60
        DO 50 IO = MN1, MNO
          C(IO,L) = ZL(L) - ZL(4)
   50   C(IO,4) = ZL(4)
```

```
  26       60 CONTINUE
        C-------------------------------------------------------------
        C
  27          SUM = 0
        C
        C FIRST (OUTERMOST) LOOP (L1) OF SIX NESTED LOOPS .
        C ITERATING THROUGH THE FOUR TERMS OF BARYCENT. COORDS.
        C EXPRESSED IN CARTES. COORDS.
        C
  28          DO 170 L1 = 1, 4
        C
        C INITIALIZE INDICES FOR XNFAC WITH AN OFFSET OF 1 :
        C
  29             N1L1 = 1
  30             N2L1 = 1
  31             N3L1 = 1
        C
        C INCREMENT INDICES FOR XNFAC :
        C
  32             IF (L1 .EQ. 1) N1L1 = N1L1 + 1
  33             IF (L1 .EQ. 2) N2L1 = N2L1 + 1
  34             IF (L1 .EQ. 3) N3L1 = N3L1 + 1
        C
        C BUILT UP COEFFS. FOR POLYNOMIAL CONSISTING OF INTEGRALS IN
        C BARYCENT. COORDS. FROM TRANSFORMATION COEFFS. IN C :
        C
  35             CMUL1 = C(1,L1)
        C
        C IF SUM OF EXPONENTS IS GT. 1 SKIP TO NEXT INNER LOOP (L2) :
        C (NO. OF USED LOOPS IS EQ. TO SUM OF POWERS)
        C
  36             IF (MNO .GT. 1) GO TO 70
        C-------------------------------------------------------------
        C SUM UP POLYNOMIAL BY USE OF COEFF. CMUL1 AND HAMMER'S
        C FORMULA :
        C
  37             SUM = SUM + CMUL1 * XNFAC(N1L1) * XNFAC(N2L1) * XNFAC(N3L1) /
             1   XNFAC(N1L1 + N2L1 + N3L1 + 1)
        C
        C SKIP TO END STATEMENT OF L1 - LOOP :
        C
  38             GO TO 170
        C-------------------------------------------------------------
        C
        C SECOND LOOP (L2) OF SIX NESTED LOOPS .
        C ITERATING THROUGH THE FOUR TERMS OF BARYCENT. COORDS.
        C EXPRESSED IN CARTES. COORDS.
        C
  39       70   DO 160 L2 = 1, 4
        C
```

```
              C TRANSFER INDICES FOR XNFAC FROM L1 - INTO  L2 - LOOP :
              C
      40                N1L2 = N1L1
      41                N2L2 = N2L1
      42                N3L2 = N3L1
              C
              C INCREMENT INDICES FOR XNFAC :
              C
      43                IF (L2 .EQ. 1) N1L2 = N1L2 + 1
      44                IF (L2 .EQ. 2) N2L2 = N2L2 + 1
      45                IF (L2 .EQ. 3) N3L2 = N3L2 + 1
              C
              C BUILT UP COEFFS. FOR POLYNOMIAL CONSISTING OF INTEGRALS IN
              C BARYCENT. COORDS. FROM TRANSFORMATION COEFFS. IN C :
              C
      46                CMUL2 = CMUL1 * C(2,L2)
              C
              C IF SUM OF EXPONENTS IS GT. 2 SKIP TO NEXT INNER LOOP (L3) :
              C (NO. OF USED LOOPS IS EQ. TO SUM OF POWERS)
              C
      47                IF (MNO .GT. 2) GO TO 80
              C
              C-----------------------------------------------------------
              C SUM UP POLYNOMIAL BY USE OF COEFF. CMUL2 AND HAMMER'S
              C FORMULA :
              C
      48                SUM = SUM + CMUL2 * XNFAC(N1L2) * XNFAC(N2L2) * XNFAC(N3L2) /
                  1     XNFAC(N1L2 + N2L2 + N3L2 + 1)
              C
              C SKIP TO END STATEMENT OF LOOP L2 :
              C
      49                GO TO 160
              C-------------------------------------------------------------
              C
              C THIRD LOOP (L3) OF SIX NESTED LOOPS .
              C ITERATING THROUGH THE FOUR TERMS OF BARYCENT. COORDS.
              C EXPRESSED IN CARTES. COORDS.
              C
      50         80     DO 150 L3 = 1, 4
              C
              C TRANSFER INDICES FOR XNFAC FROM L2 - INTO  L3 - LOOP :
              C
      51                   N1L3 = N1L2
      52                   N2L3 = N2L2
      53                   N3L3 = N3L2
              C
              C INCREMENT INDICES FOR XNFAC :
              C
      54                   IF (L3 .EQ. 1) N1L3 = N1L3 + 1
      55                   IF (L3 .EQ. 2) N2L3 = N2L3 + 1
```

```
56              IF (L3 .EQ. 3) N3L3 = N3L3 + 1
      C
      C BUILT UP COEFFS. FOR POLYNOMIAL CONSISTING OF INTEGRALS IN
      C BARYCENT. COORDS. FROM TRANSFORMATION COEFFS. IN C :
      C
57              CMUL3 = CMUL2 * C(3,L3)
      C
      C IF SUM OF EXPONENTS IS GT. 3 SKIP TO NEXT INNER LOOP (L4) :
      C (NO. OF USED LOOPS IS EQ. TO SUM OF POWERS)
      C
58              IF (MNO .GT. 3) GO TO 90
      C------------------------------------------------------------
      C SUM UP POLYNOMIAL BY USE OF COEFF. CMUL3 AND HAMMER'S
      C FORMULA :
      C
59              SUM = SUM + CMUL3 * XNFAC(N1L3) * XNFAC(N2L3) * XNFAC(N3L3)
           1      / XNFAC(N1L3 + N2L3 + N3L3 + 1)
      C
      C SKIP TO END STATEMENT OF LOOP L3 :
      C
60              GO TO 150
      C-------------------------------------------------------------
      C
      C FOURTH LOOP (L4) OF SIX NESTED LOOPS .
      C ITERATING THROUGH THE FOUR TERMS OF BARYCENT. COORDS.
      C EXPRESSED IN CARTES. COORDS.
      C
61       90     DO 140 L4 = 1, 4
      C
      C TRANSFER INDICES FOR XNFAC FROM L3 - INTO  L4 - LOOP :
      C
62                N1L4 = N1L3
63                N2L4 = N2L3
64                N3L4 = N3L3
      C
      C INCREMENT INDICES FOR XNFAC :
      C
65                IF (L4 .EQ. 1) N1L4 = N1L4 + 1
66                IF (L4 .EQ. 2) N2L4 = N2L4 + 1
67                IF (L4 .EQ. 3) N3L4 = N3L4 + 1
      C
      C BUILT UP COEFFS. FOR POLYNOMIAL CONSISTING OF INTEGRALS IN
      C BARYCENT. COORDS. FROM TRANSFORMATION COEFFS. IN C :
      C
68                CMUL4 = CMUL3 * C(4,L4)
      C
      C IF SUM OF EXPONENTS IS GT. 4 SKIP TO NEXT INNER LOOP (L5) :
      C (NO. OF USED LOOPS IS EQ. TO SUM OF POWERS)
      C
69                IF (MNO .GT. 4) GO TO 100
```

```
             C--------------------------------------------------------------
             C SUM UP POLYNOMIAL BY USE OF COEFF. CMUL4 AND HAMMER'S
             C FORMULA :
             C
    70                     SUM = SUM + CMUL4 * XNFAC(N1L4) * XNFAC(N2L4) * XNFAC(
        1                  N3L4) / XNFAC(N1L4 + N2L4 + N3L4 + 1)
             C
             C SKIP TO END STATEMENT OF LOOP L4 :
             C
    71                     GO TO 140
             C--------------------------------------------------------------
             C
             C FIFTH LOOP (L5) OF SIX NESTED LOOPS .
             C ITERATING THROUGH THE FOUR TERMS OF BARYCENT. COORDS.
             C EXPRESSED IN CARTES. COORDS.
             C
    72        100          DO 130 L5 = 1, 4
             C
             C TRANSFER INDICES FOR XNFAC FROM L4 - INTO  L5 - LOOP :
             C
    73                     N1L5 = N1L4
    74                     N2L5 = N2L4
    75                     N3L5 = N3L4
             C
             C INCREMENT INDICES FOR XNFAC :
             C
    76                     IF (L5 .EQ. 1) N1L5 = N1L5 + 1
    77                     IF (L5 .EQ. 2) N2L5 = N2L5 + 1
    78                     IF (L5 .EQ. 3) N3L5 = N3L5 + 1
             C
             C BUILT UP COEFFS. FOR POLYNOMIAL CONSISTING OF INTEGRALS IN
             C BARYCENT. COORDS. FROM TRANSFORMATION COEFFS. IN C :
             C
    79                     CMUL5 = CMUL4 * C(5,L5)
             C
             C IF SUM OF EXPONENTS IS GT. 5 SKIP TO NEXT INNER LOOP (L6) :
             C (NO. OF USED LOOPS IS EQ. TO SUM OF POWERS)
             C
    80                     IF (MNO .GT. 5) GO TO 110
             C--------------------------------------------------------------
             C SUM UP POLYNOMIAL BY USE OF COEFF. CMUL5 AND HAMMER'S
             C FORMULA :
             C
    81                     SUM = SUM + CMUL5 * XNFAC(N1L5) * XNFAC(N2L5) * XNFAC(
        1                  N3L5) / XNFAC(N1L5 + N2L5 + N3L5 + 1)
             C
             C SKIP TO END STATEMENT OF LOOP L5 :
             C
    82                     GO TO 130
             C--------------------------------------------------------------
```

```
  2 187.          C
  2 188.          C SIXTH (INNERMOST) LOOP (L6) OF SIX NESTED LOOPS .
  2 189.          C ITERATING THROUGH THE FOUR TERMS OF BARYCENT. COORDS.
  2 190.          C EXPRESSED IN CARTES. COORDS.
  2 191.          C
  2 192.      83    110          DO 120 L6 = 1, 4
  2 193.          C
  2 194.          C TRANSFER INDICES FOR XNFAC FROM L5 - INTO  L6 - LOOP :
  2 195.          C
  2 196.      84                    N1L6 = N1L5
  2 197.      85                    N2L6 = N2L5
  2 198.      86                    N3L6 = N3L5
  2 199.          C
  2 200.          C INCREMENT INDICES FOR XNFAC :
  2 201.          C
  2 202.      87                    IF (L6 .EQ. 1) N1L6 = N1L6 + 1
  2 203.      88                    IF (L6 .EQ. 2) N2L6 = N2L6 + 1
  2 204.      89                    IF (L6 .EQ. 3) N3L6 = N3L6 + 1
  2 205.          C
  2 206.          C BUILT UP COEFFS. FOR POLYNOMIAL CONSISTING OF INTEGRALS IN
  2 207.          C BARYCENT. COORDS. FROM TRANSFORMATION COEFFS. IN C :
  2 208.          C
  2 209.      90                    CMUL6 = CMUL5 * C(6,L6)
  2 210.          C
  2 211.          C SUM UP POLYNOMIAL BY USE OF COEFF. CMUL6 AND HAMMER'S
  2 212.          C FORMULA :
  2 213.          C
  2 214.      91                    SUM = SUM + CMUL6 * XNFAC(N1L6) * XNFAC(N2L6) * XNFAC(
  2 215.               1            N3L6) / XNFAC(N1L6 + N2L6 + N3L6 + 1)
  2 216.          C------------------------------------------------------------
  2 217.      92    120          CONTINUE
  2 218.      93    130        CONTINUE
  2 219.      94    140      CONTINUE
  2 220.      95    150    CONTINUE
  2 221.      96    160  CONTINUE
  2 222.      97    170 CONTINUE
  2 223.          C
  2 224.      98        QMLTI1 = SUM
  2 225.      99        GO TO 210
  2 226.          C
  2 227.          C EXITS FOR POWER SUMS MNO EQ. O OR GT.6 :
  2 228.          C
  2 229.     100    180 QMLTI1 = 1. / 6
  2 230.     101        GO TO 210
  2 231.     102    190 WRITE (6,200)
  2 232.     103    200 FORMAT (' ', 'MAX.EXPONENT OF 6 EXCEEDED')
  2 233.     104        STOP 1009
  2 234.          C
  2 235.     105    210 RETURN
  2 236.          CDS   DEBUG SUBCHK
```

350

2 237. 106 END

```
      C________________________________________________________________
      C
  1         FUNCTION QMLTI2(NO, N1, N2, N3, C1, C2, C3, IPX, IPY, IPZ)
      C
      C FUNCTION: CALCULATE THE MATRIX ELMENTS BY MULTIPLYING
      C           THE COEFFS. OF THE FINITE ELEMENTS WITH EACH
      C           OTHER AND THE APPROPRIATE INTEGRALS OF
      C           (X**M)*(Y**N)*(Z**O) OVER THE TETRAHEDRONS
      C           (I.E. OUTPUT OF QMLTI1 STORED IN QT) AND
      C           SUMMING THEM UP.
      C
      C SYMBOLS USED: C1 ,C2 ,C3  = COEFFS.OF FIN. ELEMENTS AS
      C                             DETERMINED BY SORT2 FOR THE
      C                             THE PRODUCT OF THREE POSSIBLE
      C                             FINTE ELEMENTS CONTRIBUTING TO
      C                             A MATRIX ELEMENT.
      C             IPX ,IPY ,IPZ = USER-SUPPLIED EXPONENTS OF
      C                             SPATIAL DEPENDENCIES (X , Y , Z-
      C                             DIMENSION) OF THE MEAN FLOW.
      C             IEX ,IEY ,IEZ = TECH. ARRAYS CONTROLLING PROPER
      C                             X ,Y ,Z-EXPONENT COMBINATION IN
      C                             THE THREE LOOPS ASSEMBLING THE
      C                             MATRIX ELEMENTS.
      C                        QT = ARRAY HOLDING VALUES OF INTEGRALS
      C                             OF (X**M)*(Y**N)*(Z**O) OVER THE
      C                             TETRAHEDRONS.
      C
      C
  2         REAL C1(10), C2(10), C3(10), QMLTI2, QR
  3         INTEGER NO, N1, N2, N3, IEX(10), IEY(10), IEZ(10)
  4         DATA IEX /0, 1, 0, 0, 2, 0, 0, 1, 0, 1/
  5         DATA IEY /0, 0, 1, 0, 0, 2, 0, 1, 1, 0/
  6         DATA IEZ /0, 0, 0, 1, 0, 0, 2, 0, 1, 1/
  7         COMMON /MNL/ QT(7,7,7), NL(4)
      C
  8         QMLTI2 = 0.
      C
      C 1ST LOOP (I1): VARIABLE VALUE NO FOR INITIAL ITERATION
      C IN ORDER TO REDUCE TO ONE SINGLE ITERATION FOR ANY
      C POSSIBLE EXPONENT COMBINATION (IMPORTANT FOR LIN. MATRIX
      C ELEMENTS CONSTRUCTED OF TWO FIN. ELEMENTS ONLY)
      C
  9         DO 10 I1 = NO, N1
 10           DO 10 I2 = 1, N2
 11             DO 10 I3 = 1, N3
      C
      C ASSEMBLE PROPER EXPONENT COMBINATION JEX ,JEY ,JEZ
      C ACCORDING TO DATA SETS IEX ,IEY ,IEZ :
      C
 12               JEX = IEX(I1) + IEX(I2) + IEX(I3)
 13               JEY = IEY(I1) + IEY(I2) + IEY(I3)
```

```
14                JEZ = IEZ(I1) + IEZ(I2) + IEZ(I3)
      C
      C INCREMENT OR OFFSET JEX ,JEY ,JEZ BY IPX ,IPY ,IPZ TO
      C SELECT APPROPRIATE INTEGRAL VALUE QR FROM ARRAY QT :
      C
15                QR = QT(JEX + IPX,JEY + IPY,JEZ + IPZ)
      C
      C BUILD UP MATRIX ELEMENTS (QMLTI2) BY SUMMING UP PRODUCTS
      C OF FIN. ELEMENT COEFFS. C1 ,C2 ,C3 AND INTEGRAL VALUE QR :
      C
16                QMLTI2 = QMLTI2 + C1(I1) * C2(I2) * C3(I3) * QR
      C
17       10 CONTINUE
      C
18          RETURN
      CDS   DEBUG SUBCHK
19          END
```

```
     C PROGRAM ADV
     C
     C FUNCTION: GENERATE 3D MATRIX FOR PERTURBATION INDUCED
     C           ADVECTION (I.E. ADVECTIVE MATRIX) FROM OUTPUT
     C           OF PROGRAM DSCRB AND ARRANGE ITS ELEMENTS IN
     C           CORRECT READING SEQUENCE (I.E. TRANSPOSITION).
     C
     C SUBROUTINES USED: SORT (MTS - SORTS MATRIX ELEMENTS ACCORDING
     C                          TO ONE SET OF THEIR INDICES)
     C                   VMULFF (IMSLLIB - MULTIPLIES TWO MATRICES)
     C
     C LOG.UNITS ADDRESSED: 3 : WORK - ELEMENTS SORTED AFTER 1ST COLUMN
     C                      4 : WORK - ELEMENTS SORTED AFTER 2ND COLUMN
     C                      9 : INPUT - ALHI
     C                     13 : INPUT - NONLIN. ADVECTIVE MATRIX FROM
     C                                  PROGRAM DSCRB
     C                     14 : WORK - NONLIN. ADVECTIVE MATRIX
     C                                 SORTED ACCORDING TO 2D SECTIONS
     C                     15 : OUTPUT - NONLIN. ADVECTIVE MATRIX
     C                                 TRANSPOSED INTO READING SEQUENCE
     C
     C SYMBOLS USED: A    = ARRAY FOR 2D SECTIONS OF THE 3D MATRIX
     C               ALHI = INVERSE OF LHS OF ODE-SYSTEM (OUTPUT
     C                      OF DSCRB)
     C               AC   = ARRAY HOLDING ALHI*A
     C               AT   = TECH. ARRAY HOLDING ROWS OF AC
     C               BT   = ARRAY HOLDING TRANSPOSE OF AC
     C
1          INTEGER L /12/, M /12/, N /12/, IA /12/, ILHI /12/, IAC /12/, IER
2          REAL A(12,12), ALHI(12,12), AC(12,12), AT(12), BT(12,12)
     C
     C READ ALHI FROM LOG.UNIT 9 :
     C
3          DO 10 IRL = 1, ILHI
4       10 READ (9) (ALHI(ICL,IRL),ICL=1,ILHI)
     C
     C SORT 3D MATRIX ELEMENTS READ FROM LOG.UNIT 13 IN ASCENDING
     C ORDER BY THEIR TWO COLUMN NOS. INTO FILES -3S ,-4S :
     C
5          CALL SORT('S=FI,A,5,4 I=13,VB,16,24 O=-3S,VB,16,24 R=650 E ', &
         1      120)
6          CALL SORT('S=FI,A,9,4 I=13,VB,16,24 O=-4S,VB,16,24 R=650 E ', &
         1      120)
     C
     C------ ASSEMBLE 2D SECTIONS OF 3D MATRIX IN ARRAY A :--------
     C
     C LOOP ICC ITERATES THROUGH SORTED INDICES (I.E. I1 ,I2) :
     C
7          DO 70 ICC = 1, N
     C
```

```
C CLEAR ARRAY A FROM PREV. D2 SECTION :
C
       DO 20 IR = 1, L
         DO 20 IC = 1, M
   20  A(IC,IR) = O.
C
C READ ELEMENTS AND INDICES SORTED AFTER 1ST COLUMN NO. I1
C FROM LOG.UNIT 3 ATTACHED TO FILE -3S INTO A :
C
   30  READ (3,END=40) EL, I1, I2, IR
       IF (I1 .NE. ICC) GO TO 40
       A(I2,IR) = A(I2,IR) + EL
       GO TO 30
C
C READ ELEMENTS AND INDICES SORTED AFTER 2ND COLUMN NO. I2
C FROM LOG.UNIT 4 ATTACHED TO FILE -4S INTO A :
C
   40  READ (4,END=50) EL, I1, I2, IR
       IF (I2 .NE. ICC) GO TO 50
       A(I1,IR) = A(I1,IR) + EL
       GO TO 40
C
C MULTIPLY 2D SECTIONS (A) WITH ALHI AND STORE IN AC :
C
   50  CALL VMULFF(A, ALHI, L, M, N, IA, ILHI, AC, IAC, IER)
       IF (IER .NE. O) STOP 90
C
C WRITE MATRICES FROM ARRAY AC INTO LOG.UNIT 14 :
C
       DO 60 IRR = 1, IAC
   60  WRITE (14) (AC(IC,IRR),IC=1,N)
C
   70 CONTINUE
C
C-------- TRANSPOSING SECTION ------------------------------
C
C JR-LOOP ITREATES THROUGH ROWS PER MATRIX STORED IN LOG.UNIT 14 :
C
      DO 110 JR = 1, IAC
C
C K-LOOP ITERATES THROUGH ALL THE MATRICES STORED IN LOG.UNIT 14 :
C
        DO 90 K = 1, M
C
C CALCULATE WHICH LINE IN LOG.UNIT 14 HOLDS THE CURRENT ROW
C OF THE CURRENT MATRIX AND READ THIS LINE INTO AT :
C
          LINE = (JR + (K - 1)*IAC) * 1000
          READ (14'LINE) AT
C
```

```
              C FILL UP BT :
              C
        28              DO 80 IC = 1, IAC
        29       80     BT(K,IC) = AT(IC)
              C
        30       90   CONTINUE
              C
              C WRITE BT INTO LOG.UNIT 15 :
              C
        31              DO 100 IC = 1, IAC
        32      100   WRITE (15) (BT(K,IC),K=1,M)
              C
        33      110 CONTINUE
              C
        34      120 STOP
              CDS    DEBUG SUBCHK
        35           END
```

```
C PROGRAM BIF3A
C
C MAIN LINE FOR BIFURCATION SEQUENCE (NS-DISCRETIZATION, 12-DIM.
C                                     ,DOUBLE PRECISION)
C
C FUNCTION: TEST ODE-SYSTEM REPRESENTING THE NS-EQUATION FOR
C           A PERTURBATION SUPERIMPOSED ON A STEADY MEAN
C           (BASIC) FLOW FOR HOPF BIFURCATONS. EVALUATE AN
C           ANALYT. APPROXIMATION FOR HOPF ORBITS (AS GIVEN
C           BY HASSAD AND KAZARINOFF WITH THE SUBROUTINE BIFOR2)
C           AND TEST IT AT FIXED TIMES (I.E. IN A LOOP WITH
C           DISCRETE TIMESTEPS) FOR SECONDARY HOPF BIFURCATONS.
C           EVALUATE THE ANALYT. APPROXIMATIONS FOR THEIR HOPF
C           ORBITS TO CONSTRUCT A COORD.SYSTEM FOR THE ASSOC-
C           IATED 2-TORUS AND TO TEST THESE SECONDARY HOPF
C           ORBITS AT FIXED TIMES FOR 3RD HOPF BIFURCATIONS
C           EVALUATE THE ANALYT. APPROXIMATIONS FOR THEIR
C           HOPF ORBITS TO CONSTRUCT A COORD.SYSTEM FOR THE
C           ASSOCIATED 3-TORUS.
C
C SUBROUTINES USED: BIFOR2 (THIS ROUTINE PERFORMS THE ACTUAL
C                   TEST FOR HOPF BIFURCATION AND CALCULATES
C                   ALL ASSOCIATED PARAMETERS SUCH AS THE CRIT.
C                   VALUE OF THE BIFURCATION PARAMETER AND
C                   PARAMETERS USED IN THE EVALUATION OF THE
C                   ANALYT. APPROXIMATION OF THE HOPF ORBITS.
C                   AUTHOR OF BIFOR2: B. HASSARD, DEPT. OF
C                   MATHEMATICS, SUNY AT BUFFALO, BUFFALO N.Y.
C                   14214)
C                   FUN  (EVALUATES ODE-SYSTEM AND ITS JACOBIAN
C                         FOR 1ST PERTURBATION)
C                   FUNP (EVALUATES ODE-SYSTEM AND ITS JACOBIAN
C                         FOR 2ND AND 3RD PERTURBATION)
C                   MFC  (EVALUATES MEAN (BASIC) FLOW AT CRIT.
C                         VALUE OF 1ST BIFURCATION)
C
C LOG.UNITS ADDRESSED:  1 : OUTPUT - ANUP(1) ,PAR(2) ,V1(1)
C                       2 : OUTPUT - ANUP(2) ,PAR(2) ,V1(2)
C                       3 : OUTPUT - ANUP(3) ,PAR(3) ,V1(3)
C                       4 : INPUT  - CONTROL PARAMETERS
C                       6 : OUTPUT - ERROR MESSAGES - TERMINAL
C                       7 : INPUT  - IC1
C                       8 : INPUT  - IC2
C                       9 : WORK   - VS
C                      10 : WORK   - ACIR
C                      11 : INPUT  - ACIN
C                      12 : INPUT  - ANL1
C                      13 : OUTPUT - ABLOC ,ABLOT
C                      15 : INPUT  - A3D
C                      17 : WORK   - ACIT
```

```
C                        19 : WORK    - A3T
C
C SYMBOLS USED: N     = DIMENSION OF ODE-SYSTEM (I.E.12)
C               NN    = COLUMN NO. OF ITS NONLIN. MATRIX (I.E.78)
C               NCF   = NO. OF LIN. MATRICES (I.E.13)
C               V     = ARRAY HOLDING 3 SETS OF PERTURBATION
C                       VELOCITIES OF THE 3 BIFURCATIONS
C               VI    = INIT. GUESSES FOR CRITICAL PERTURBATION
C                       VELOCITIES (ALL KEPT IDENT. ZERO)
C               ANL1  = NONLIN. MATRIX OF SYSTEM
C              ABLOC  = LIN. MATRIX OF SYSTEM AT CRIT. VALUE
C                       OF BIFURCATION PARAMETER ANUC
C              ABLOT  = TRANSPOSE OF ABLOC
C               A1IN  = SINGLE PRECISION ARRAY HOLDING 1 ROW
C                       OF NONLIN. MATRIX ANL1
C               ACIN  = SINGLE PRECISION ARRAY HOLDING 1 ELE-
C                       MENT FROM EACH ONE OF THE 13 LIN.
C                       MATRICES
C               A3D   = ARRAY HOLDING 1 ELEMENT FROM EACH OF
C                       THE 12 ADDITIONAL MATRICES FOR THE
C                       ADVECTIVE TERMS STEMMING FROM PERTURB-
C                       ATION VELOCITIES
C               A3T   = SINGLE PRECISION ARRAY HOLDING ELEMENTS
C                       TRANSPOSE TO THE ONES IN A3D
C               ACIR  = SINGLE PRECISION ARRAY HOLDING 1 ELE-
C                       MENT FROM EACH LIN. MATRIX USED FROM
C                       THE TOTAL OF 13 LIN. MATRICES
C               ACIT  = SINGLE PRECISION ARRAY HOLDING ELEMENTS
C                       TRANSPOSE TO THE ONES IN ACIR
C               V1    = COMPLEX EIGENVECTOR OF THE JACOBIAN
C                       MATRIX AT THE CRIT. VALUE OF THE BIFURC-
C                       ATION PARAMETER ANU - USED IN EVALUATION
C                       OF ANALYTIC APPROXIMATIONS FOR PERTURB-
C                       ATION VELOCITIES (I.E. HOPF ORBITS) VP.
C                       (OUTPUT OF BIFOR2)
C                 VP  = ARRAY HOLDING PERTURBATION VELOCITIES
C                PAR  = PARAMETERS FOR EVALUATION OF ANALYT.
C                       APPROXIMATION OF PERTURBATION VELOCITIES
C                       (I.E. HOPF ORBITS) VP. (OUTPUT OF BIFOR2)
C               ANUP  = USER-SUPPLIED INIT. GUESSES FOR THE CRIT.
C                       VALUES OF BIFURCATION PARAMETER -
C                       UPON RETURN: CRIT. VALUE OF IT
C               VS    = TECH. ARRAY FOR TRANSFER OF 2ND PERTUB-
C                       ATION VELOCITY
C               VC    = MEAN (BASIC) FLOW AT CRIT. VALUE OF 1ST
C                       BIFURCATION
C          IC1 ,IC2   = INDEX CONTROL ARRAYS TO CORRELATE
C                       PRODUCTS OF ELEMENTS OF V WITH THE
C                       CORRECT COLUMN NO. OF ANL1
C               CF    = COEFFS. OF LIN. MATRICES - I.E. PHYS.
```

```
     C                       FLOW PARAMETERS (SHEAR COEFFS. ,ABSOLUTE
     C                       COEFFS. ,VISCOSITY)
     C                LCF  = SWITCHES FOR CF
     C                IP   = INDICATOR FOR 2ND (=1) OR 3RD (=2)
     C                       BIFURCATION (USED BY FUNP)
     C
     C REMARK: TECH. PARAMETERS FOR BIFOR2 ARE NOT LISTED HERE.
     C         AS FAR AS THEY ARE USER-SUPPLIED ,THEY ARE LISTED
     C         IN THE CONTROL PARAMETER FILE 4.NS.A ATTACHED TO
     C         LOG.UNIT 4. (SEE ALSO DOCUMENTATION FOR BIFOR2)
     C
   1       IMPLICIT REAL*8(A - H,O - Z)
   2       INTEGER N, NN, NCF, MTH1, MTH2, MTH3, JJOB, IPRINT, ICK(3), IG
   3       LOGICAL*1 LFMT(1) /'*'/
   4       REAL*8 V(12,3), VI(3,3), CFT(13)
   5       REAL*8 ERR(7), W(432)
     C*****************************************************************
     C* NB: SET DIM. OF ACIR,ACIT(LINE BELOW) = # OF COEFFS.(CF,CFN) *
   6       REAL*4 A1IN(78), ACIN(13), A3T(12), ACIR(7), ACIT(7)
   7       COMPLEX*16 V1(12,3)
     C
   8       COMMON /CBL/ ANL1(78,12), CF(13), ABLOC(12,12), ABLOT(12,12), V1,
          1       VP(12), PAR(10,3), ANUP(3), CP(2), CC(10), IX(10), VS(12),
          2       IC1(78), IC2(78), IS(2), IT(2), NT(2), NN, ICR, IP
   9       COMMON /LF/ LCF(13)
  10       COMMON /CBT/ CFT
     C
  11       N = 12
  12       NN = 78
  13       NCF = 13
     C
  14       EXTERNAL FUN, FUNP
     C
     C-----------------------------------------------------------------
     C READ USER-SUPPLIED TECH. AND PHYS. CONTROL PARAMETERS AND SWITCHES
     C FROM LOG.UNIT 4 ATTACHED TO FILE 4.NS.A :
     C
  15       READ (4'3000,LFMT) LCF
  16       READ (4'4000,LFMT) CF
  17       READ (4'6000,LFMT) CC
  18       READ (4'8000,LFMT) IX
  19       READ (4'10000,LFMT) ANUP
  20       READ (4'12000,LFMT) IA12, IA23
  21       READ (4'14000,LFMT) DA12, DA23
  22       READ (4'16000,LFMT) VI
     C
  23       READ (4'20000,LFMT) CP
  24       READ (4'22000,LFMT) NT
  25       READ (4'24000,LFMT) IS
  26       READ (4'26000,LFMT) NO
```

```
 27         READ (4'28000,LFMT) IR1, IR2, IR3
 28         READ (4'30000,LFMT) IG1, IG2
 29         READ (4'32000,LFMT) TI1, TI2
      C
 30         READ (4'36000,LFMT) U1, U2, U3
 31         READ (4'38000,LFMT) EPSR1, EPSR2, EPSR3
 32         READ (4'40000,LFMT) EPS1, EPS2, EPS3
 33         READ (4'42000,LFMT) NSIG1, NSIG2, NSIG3
 34         READ (4'44000,LFMT) ITMAX
 35         READ (4'46000,LFMT) ICK
 36         READ (4'48000,LFMT) MTH1, MTH2, MTH3
 37         READ (4'50000,LFMT) JJOB1, JJOB2, JJOB3
 38         READ (4'52000,LFMT) IPRNT1, IPRNT2, IPRNT3
      C
      C-----------------------------------------------------------------
      C THE SWITCHES IR1 ,IR2 ,IR3 ALLOW TO USE CRIT. VALUES OF PREVIOUS
      C BIFURCATIONS AS INIT. GUESSES (USER - OPTION)
      C
 39         IF (IR1 .EQ. 1) READ (1) ANUP(1)
 40         IF (IR2 .EQ. 1) READ (2) ANUP(2)
 41         IF (IR3 .EQ. 1) READ (3) ANUP(3)
 42         IF (IR1 .EQ. 1) REWIND 1
 43         IF (IR2 .EQ. 1) REWIND 2
 44         IF (IR3 .EQ. 1) REWIND 3
      C
      C----------------------------------------------------------------
      C READ INDEX CONTROL ARRAYS IC1 ,IC2 :
      C (OUTPUT FROM DISCRETIZATION PROGRAM DSCRB)
      C
 45         READ (7) IC1
 46         READ (8) IC2
      C
      C READ LIN. (ACIN) AND NONLIN. MATRICES (ANL1) :
      C (OUTPUT FROM DISCRETIZATION PROGRAM DSCRB)
      C
 47         DO 30 IR = 1, N
 48           READ (12) A1IN
 49           DO 20 IC = 1, NN
      C
      C COPY FROM SINGLE TO DOUBLE PRECISION ARRAY :
      C
 50             ANL1(IC,IR) = -A1IN(IC)
      C
      C SKIP READ OF LIN. MATRICES IF 2ND AND/OR 3RD BIFURCATIONS
      C ARE TO BE PERFORMED ONLY (USER-OPTION) :
      C
 51             IF (IG1 .EQ. 1 .OR. IG2 .EQ. 1 .OR. IC .GT. N) GO TO 20
      C
 52             READ (11) ACIN
 53             ICR = 0
```

```
              DO 10 ICF = 1, NCF
C
C SELECT LIN. MATRICES USED IN CURRENT MEAN (BASIC) FLOW
C COPY FROM SINGLE TO DOUBLE PRECISION ARRAY ACIR :
C
                IF (LCF(ICF) .EQ. 0) GO TO 10
                ICR = ICR + 1
                ACIR(ICR) = -ACIN(ICF)
   10         CONTINUE
C
C STORE SELECTED LIN. MATRICES IN LOG.UNIT 10 (USED IN
C SUBROUTINE FUN) :
C
              WRITE (10) (ACIR(I),I=1,ICR)
C
   20       CONTINUE
   30 CONTINUE
C
C SKIP TRANSPOSITION OF LIN. MATRICES IF 2ND AND/OR 3RD
C BIFURCATIONS ARE TO BE PERFORMED ONLY (USER-OPTION) :
C
        IF (IG1 .EQ. 1 .OR. IG2 .EQ. 1) GO TO 80
C
C-----------------------------------------------------------------
C COPY TRANSPOSES OF SELECTED LIN. MATRICES AND ADVECTIVE
C MATRICES INTO LOG.UNITS 17 AND 19 (USED IN JACOBIAN EVALUAT-
C ION IN SUBROUTINES FUN AND FUNP) :
C
        DO 50 ITR = 1, N
          DO 40 ITC = 1, N
            LINE = (ITR + (ITC - 1)*N) * 1000
            READ (10'LINE) ACIT
            WRITE (17) (ACIT(I),I=1,ICR)
   40     CONTINUE
   50 CONTINUE
C
        DO 70 JTR = 1, N
          DO 60 JTC = 1, N
            LINE = (JTR + (JTC - 1)*N) * 1000
            READ (15'LINE) A3T
            WRITE (19) (A3T(I),I=1,N)
   60     CONTINUE
   70 CONTINUE
C
C-----------------------------------------------------------------
C ECHO USER-SUPPLIED CONTROL PARAMETERS :
C
   80 WRITE (6,LFMT) LCF
      WRITE (6,LFMT) CF
      WRITE (6,LFMT) CC
      WRITE (6,LFMT) IX
      WRITE (6,290) ANUP
      WRITE (6,300) IA12, IA23
      WRITE (6,310) DA12, DA23
      WRITE (6,320) VI
C
      WRITE (6,330) CP
      WRITE (6,340) NT
      WRITE (6,350) IS
      WRITE (6,360) NO
      WRITE (6,370) IR1, IR2, IR3
      WRITE (6,380) IG1, IG2
      WRITE (6,390) TI1, TI2
C
      WRITE (6,400) U1, U2, U3
      WRITE (6,410) EPSR1, EPSR2, EPSR3
      WRITE (6,420) EPS1, EPS2, EPS3
      WRITE (6,430) NSIG1, NSIG2, NSIG3
      WRITE (6,440) ITMAX
      WRITE (6,450) ICK
      WRITE (6,460) MTH1, MTH2, MTH3
      WRITE (6,470) JJOB1, JJOB2, JJOB3
      WRITE (6,480) IPRNT1, IPRNT2, IPRNT3
C
C-----------------------------------------------------------
C TRANSFER AND ECHO COMPONENTS OF INIT. GUESS VELOCITIES :
C
      DO 100 J = 1, 3
        DO 90 IN = 1, N, 3
          V(IN,J) = VI(1,J)
          V(IN + 1,J) = VI(2,J)
          V(IN + 2,J) = VI(3,J)
   90   CONTINUE
  100 CONTINUE
C
  110 WRITE (6,120) ((V(I,J),J=1,3),I=1,N)
C
  120 FORMAT ('O INIT.GUESS VELOCITIES', (1X,3E20.10))
      WRITE (6,130)
  130 FORMAT (1X)
C
C---------------------------------------------------------
C
C --- BIFURCATION SEQUENCES AND ASSOCIATED TIME LOOPS -----
C
      IER = 0
C
C TEST MEAN FLOW FOR 1ST BIFURCATION OR SKIP IF ALREADY PER-
C FORMED IN PREVIOUS RUN :
C
```

```
      IF (IG1 .EQ. 1) GO TO 140
      CALL BIFOR2(FUN, V(1,1), N, ANUP(1), U1, MTH1, JJOB1, IPRNT1,
     1     PAR(1,1), V1(1,1), ERR, W, IER, EPSR1, EPS1, NSIG1, ITMAX,
     2     ICK(1))
      IF (IER .NE. 0) STOP 1
      ICK(1) = 0
C
C WRITE OUTPUT PARAMETERS OF 1ST BIFURCATION AND VALUES OF
C LIN. MATRICES AT CRIT. POINT (ABLOC ,ABLOT) ON LOG.UNITS 1 ,13 :
C
      WRITE (1) ANUP(1), (PAR(IP1,1),IP1=1,10), (V1(IV1,1),IV1=1,N)
      WRITE (13) ABLOC, ABLOT
C
C EVALUATE MEAN FLOW AT CRIT. POINT :
C
      CALL MFC
C
C STOP IF ONLY 1ST BIFURCATION IS TO BE PERFORMED (USER-OPTION) :
C
      IF (NO .EQ. 1) STOP 11
      GO TO 210
C-----------------------------------------------------------------
C IF 1ST BIFURCATION WAS DONE IN PREVIOUS RUN ,READ AND ECHO ITS
C OUTPUT PARAMETERS AND ABLOC ,ABLOT FROM LOG.UNITS 1 AND 13 :
C
  140 READ (1) ANUP(1), (PAR(JP1,1),JP1=1,10), (V1(JV1,1),JV1=1,N)
      READ (13) ABLOC, ABLOT
C
      WRITE (6,150) ANUP(1)
  150 FORMAT ('CRIT.ANUP : ', 3X, E20.10)
      WRITE (6,160) (PAR(I,1),I=1,5)
  160 FORMAT (/26X, 4HAMU2, 16X, 4HTAU2, 16X, 5HBETA2, 14X, 9HRE(C1(0)),
     1        11X, 9HIM(C1(0)), /8H  VALUES, 9X, 5E20.10)
      WRITE (6,170) (PAR(I,1),I=6,10)
  170 FORMAT (/25X, 6HDALPHA, 14X, 6HDOMEGA, 14X, 8HOMEGA(0), 12X,
     1        6HENORMX, 14X, 6HENORMV, /8H  VALUES, 9X, 5E20.10)
      WRITE (6,180)
  180 FORMAT (/14X, 17H|  EIGENVECTOR V1)
      WRITE (6,190) (I,V1(I,1),I=1,N)
  190 FORMAT (8H   |   V1(, I2, 3H) =, 2E20.10)
      WRITE (6,200)
  200 FORMAT (1X)
C
C-----------------------------------------------------------------
C TIME LOOP FOR 2ND BIFURCATIONS OVER 1ST ORBIT :
C
C INITIALIZE NO. OF TIME STEPS OR ORBIT DIVISIONS AND INIT.
C GUESS FOR ANUP(2) BY (USER-OPTION) INCREMENTING 1ST CRIT.VALUE :
C
  210 NT1 = NT(1)
```

```
      IF (IA12 .EQ. 1) ANUP(2) = ANUP(1) + DA12
C
      DO 250 IT1 = 1, NT1
        IP = 1
        IT(1) = IT1
C
C TEST FOR 2ND BIFURCATION OR SKIP IF ALREADY PERFORMED IN
C PREVIOUS RUN :
C
        IF (IG2 .EQ. 1) GO TO 220
        CALL BIFOR2(FUNP, V(1,2), N, ANUP(2), U2, MTH2, JJOB2, IPRNT2,
     1       PAR(1,2), V1(1,2), ERR, W, IER, EPSR2, EPS2, NSIG2, ITMAX,
     2       ICK(2))
        IF (IER .NE. 0) STOP 2
        ICK(2) = 0
C
C WRITE OUTPUT PARAMETERS ON LOG.UNIT 2 AND CURRENT VALUE OF
C 1ST HOPF ORBIT (VS) ON LOG.UNIT 9
C
        WRITE (2) ANUP(2), (PAR(IP2,2),IP2=1,10), (V1(IV2,2),IV2=1,N)
        WRITE (9) VS
C
C STOP IF ONLY 1ST AND/OR 2ND BIFURCATIONS ARE TO BE PERFORMED
C (USER-OPTION) :
C
        IF (NO .EQ. 2) GO TO 250
C
        GO TO 230
C--------------------------------------------------------------
C IF 2ND BIFURCATION WAS DONE IN PREVIOUS RUN ,READ (AND ECHO)
C ITS OUTPUT PARAMETERS AND VS FROM LOG.UNITS 2 AND 9 :
C
  220   READ (2) ANUP(2), (PAR(JP2,2),JP2=1,10), (V1(JV2,2),JV2=1,N)
        READ (9) VS
C
        WRITE (6,150) ANUP(2)
        WRITE (6,160) (PAR(I,2),I=1,5)
        WRITE (6,170) (PAR(I,2),I=6,10)
        WRITE (6,180)
        WRITE (6,190) (I,V1(I,2),I=1,N)
        WRITE (6,200)
C
C--------------------------------------------------------------
C TIME LOOP FOR 3RD BIFURCATIONS OVER 2ND ORBITS :
C
C INITIALIZE NO. OF TIME STEPS OR ORBIT DIVISIONS AND INIT.
C GUESS FOR ANUP(3) BY (USER-OPTION) INCREMENTING 2ND CRIT.VALUE :
C
  230   IP = 2
        NT2 = NT(2)
```

```
 159          IF (IA23 .EQ. 1) ANUP(3) = ANUP(2) + DA23
       C
 160          DO 240 IT2 = 1, NT2
 161            IT(2) = IT2
 162            CALL BIFOR2(FUNP, V(1,3), N, ANUP(3), U3, MTH3, JJOB3, IPRNT3,
            1        PAR(1,3), V1(1,3), ERR, W, IER, EPSR3, EPS3, NSIG3,
            2        ITMAX, ICK(3))
 163            IF (IER .NE. 0) STOP 3
 164            ICK(3) = 0
       C
       C WRITE OUTPUT PARAMETERS ON LOG.UNIT 3 :
       C
 165            WRITE (3) ANUP(3), (PAR(IP3,3),IP3=1,10), (V1(IV3,3),IV3=1,N)
       C
       C-----------------------------------------------------------------
       C
 166     240   CONTINUE
 167     250 CONTINUE
       C
       C-----------------------------------------------------------------
       C
 168     260 FORMAT ('IER =', 2X, I3)
 169     270 FORMAT (/, 10(3X,F7.3))
 170     280 FORMAT (/, 7(3X,F7.3))
 171     290 FORMAT (' ANU = ', 3(G20.10,2X))
 172     300 FORMAT (' IA12 = ', I1, 3X, 'IA23 = ', I1)
 173     310 FORMAT (' DA12 = ', G10.5, 3X, 'DA23 = ', G10.5)
 174     320 FORMAT (' VI1=', 3(G10.5), 'VI2=', 3(G10.5), 'VI3=', 3(G10.5))
        C
 175     330 FORMAT (' CP(1) = ', F7.5, 2X, 'CP(2) = ', F7.5)
 176     340 FORMAT (' NT(1) = ', I3, 2X, 'NT(2) = ', I3)
 177     350 FORMAT (' IS(1) = ', I3, 2X, 'IS(2) = ', I3)
 178     360 FORMAT (' NO = ', I1)
 179     370 FORMAT (' IR1 = ', I1, 3X, 'IR2 = ', I1, 3X, 'IR3 = ', I1)
 180     380 FORMAT (' IG1 = ', I1, 3X, 'IG2 = ', I1)
 181     390 FORMAT (' TI1 = ', G20.10, 3X, 'TI2 = ', G20.10)
        C
 182     400 FORMAT ('     U = ', 3(G20.10,2X))
 183     410 FORMAT ('  EPSR = ', 3(G20.10,2X))
 184     420 FORMAT ('   EPS = ', 3(G20.10,2X))
 185     430 FORMAT ('  NSIG = ', 3(I2,2X))
 186     440 FORMAT (' ITMAX = ', I2)
 187     450 FORMAT ('   ICK = ', 3(I1,2X))
 188     460 FORMAT ('   MTH = ', 3(I1;2X))
 189     470 FORMAT ('  JJOB = ', 3(I1,2X))
 190     480 FORMAT ('IPRINT = ', 3(I1,2X))
        C
 191          STOP
        CDS   DEBUG SUBCHK
 192          END
```

```
      C
      C==========================================================
      C
1           SUBROUTINE FUN(V, N, ANU, F, DF, IND)
      C
      C FUNCTION: ASSEMBLE ODE-SYSTEM REPRESENTING THE DISCRETIZ-
      C           ATION OF THE EQUATION FOR 1ST ORDER PERTURBAT-
      C           IONS ON THE NS-EQUATION. THE BIFURCATION PAR-
      C           AMETER ANU IS INTRODUCED BY THE MEAN FLOW THROUGH
      C           ITS ADVECTIVE TERMS.
      C
      C LOG.UNITS ADDRESSED: 10 : WORK   - ACIR
      C                      17 : WORK   - ACIT
      C
      C SYMBOLS USED: N     = DIMENSION OF ODE-SYSTEM (I.E.12)
      C               NN    = COLUMN NO. OF ITS NONLIN. MATRIX (I.E.78)
      C               V     = ARRAY HOLDING 3 SETS OF PERTURBATION
      C                       VELOCITIES OF THE 3 BIFURCATIONS
      C               F     = LHS OF ODE-SYSTEM TO BE EVALUATED
      C               DF    = JACOBIAN OF ODE-SYSTEM TO BE EVALUATED
      C               CFT   = TECH. ARRAY FOR TRANSFER OF CRIT.
      C                       COEFFS. OF LIN. MATRICES FROM SUB-
      C                       ROUTINE FUN INTO SUBROUTINE MFC
      C               ANL1  = NONLIN. MATRIX OF SYSTEM
      C               ABLOC = LIN. MATRIX OF SYSTEM AT CRIT. VALUE
      C                       OF BIFURCATION PARAMETER ANUC
      C               ABLOT = TRANSPOSE OF ABLOC
      C               ACIR  = SINGLE PRECISION ARRAY HOLDING 1 ELE-
      C                       MENT FROM EACH LIN. MATRIX USED FROM
      C                       THE TOTAL OF 13 LIN. MATRICES
      C               ACIT  = SINGLE PRECISION ARRAY HOLDING ELEMENTS
      C                       TRANSPOSE TO THE ONES IN ACIR
      C               V1    = COMPLEX EIGENVECTOR OF THE JACOBIAN
      C                       MATRIX AT THE CRIT. VALUE OF THE BIFURC-
      C                       ATION PARAMETER ANU - USED IN EVALUATION
      C                       OF ANALYTIC APPROXIMATIONS FOR PERTURB-
      C                       ATION VELOCITIES (I.E. HOPF ORBITS) VP.
      C                       (OUTPUT OF BIFOR2)
      C         CC ,CF ,IX  = USER-SUPPLIED FACTORS ,COEFFS. AND EXPONENTS
      C                       USED TO CONSTRUCT LINEAR ,AFFINE OR POWER
      C                       DEPENDENCIES OF PHYSICAL PARAMETERS OF
      C                       THE MEAN (BASIC) FLOW (I.E. SHEAR
      C                       ,VISCOSITY) ON BIFURCATION PARAMETER ANUP
      C               VP    = ARRAY HOLDING PERTURBATION VELOCITIES
      C               PAR   = PARAMETERS FOR EVALUATION OF ANALYT.
      C                       APPROXIMATION OF PERTURBATION VELOCITIES
      C                       (I.E. HOPF ORBITS) VP. (OUTPUT OF BIFOR2)
      C               ANU   = BIFURCATION PARAMETER. UPON ENTRY: INIT.
      C                       GUESS - UPON RETURN CRIT. VALUE
      C           IC1 ,IC2  = INDEX CONTROL ARRAYS TO CORRELATE
```

```
C                      PRODUCTS OF ELEMENTS OF V WITH THE
C                      CORRECT COLUMN NO. OF ANL1
C               CFN  = COEFFS. OF LIN. MATRICES - I.E. PHYS.
C                      FLOW PARAMETERS (SHEAR COEFFS. ,ABSOLUTE
C                      COEFFS. ,VISCOSITY)
C               IND  = SWITCH CONTROLLING EVALUATION OF ODE-
C                      SYSTEM OR ITS JACOBIAN (USED WITHIN
C                      BIFOR2 ONLY)
C               ICT  = SWITCH CONTROLLING TRANSFER AND
C                      STORAGE OF COEFFS. AND LIN. MATRICES
C                      AT CRIT. ANU FOR LATER USE IN SUB-
C                      ROUTINES FUNP AND MFC (SET WITHIN
C                      BIFOR2)
C
      IMPLICIT REAL*8(A - H,O - Z)
      INTEGER N
      REAL*8 V(N), F(N), DF(N,N), CFN(13), CFT(13)
      COMPLEX*16 V1(12,3)
C******************************************************************
C* NB: SET DIM. OF ACIR,ACIT(LINE BELOW) = # OF COEFFS.(CF,CFN) *
      REAL*4 ACIR(7), ACIT(7)
      COMMON /CBL/ ANL1(78,12), CF(13), ABLOC(12,12), ABLOT(12,12), V1,
     1       VP(12), PAR(10,3), ANUP(3), CP(2), CC(10), IX(10), VS(12),
     2       IC1(78), IC2(78), IS(2), IT(2), NT(2), NN, ICR, IP
      COMMON /CT/ ICT
      COMMON /CBT/ CFT
C
C***** SETTING OF FLOW STRUCTURE *******************************
C DONE BY USER WITHIN THIS PROGRAM SECTION :
C SET DEPENDENCE OF MEAN FLOW ON ANU BY COOSING FUNCTIONAL
C DEPENDENCIES (LINEAR ,POWER ,AFFINE) BETWEEN THE FLOW COEFFS.
C CFN AND ANU. USED COEFFS. CF MUST BE IN ASCENDING ORDER W.R.T
C THEIR INDICES (WHICH DENOTE THE LIN. MATRIX THEY ARE ASSOC.
C WITH) AND UNUSED ONES DELETED (I.E. COMMENTED OUT).
C THE INDICES OF THE COEFFS. CF IDENTIFY WITH THE PHYS. FLOW
C COEFFS. AS SET IN LINES 3 (LCF) AND 4 (CF) OF THE CONTROL
C PARAMETER FILE (LOG.UNIT 4) :
C
C       1 ¬ C( X)         2 ¬ C( Y)     3 ¬ C( Z)    4 ¬ VISCOSITY
C       5 ¬ C(XX)         6 ¬ C(XY)     7 ¬ C(XZ)
C       8 ¬ C(YX)         9 ¬ C(YY)    10 ¬ C(YZ)
C      11 ¬ C(ZX)        12 ¬ C(ZY)    13 ¬ C(ZZ)
C
       CFN(1) = CF(1) * ANU
       CFN(2) = CF(2) * ANU
       CFN(3) = CF(3) * ANU
       CFN(4) = -CF(4)
       CFN(5) = CF(6) * ANU + CC(5)
       CFN(6) = CF(10) * ANU + CC(6)
       CFN(7) = CF(11) * ANU + CC(7)
```

```
CC    CFN( 8) =
CC    CFN( 9) =
CC    CFN(10) =
CC    CFN(11) =
CC    CFN(12) =
CC    CFN(13) =
C*************************************************************
C
C COPY CRIT. VALUES OF COEFFS. CFN INTO TRANSFER ARRAYS CFT
C (FOR USE IN MFC) :
C
      IF (ICT .NE. 1) GO TO 20
      DO 10 IK = 1, ICR
   10 CFT(IK) = CFN(IK)
C
   20 IF (IND .EQ. 1) GO TO 90
C
C --- FUNCTION EVALUATION --------------
C
      REWIND 10
C
C ROW-LOOP IR :
C
      DO 80 IR = 1, N
         FS = 0.DO
C
C COLUMN-LOOP IC :
C
        DO 70 IC = 1, NN
C
C EVALUATE NONLIN. MATRIX :
C
          I1 = IC1(IC)
          I2 = IC2(IC)
          FINCR = ANL1(IC,IR) * V(I1) * V(I2)
          IF (IC .GT. N) GO TO 60
C
C ASSEMBLE LIN. MATRIX :
C
   30     READ (10) ACIR
          ABLC = O.DO
          DO 40 ICF = 1, ICR
            ACMB = ACIR(ICF)
            ABLC = ABLC + CFN(ICF) * ACMB
   40     CONTINUE
C
C COPY CRIT. VALUE OF LIN. MATRIX INTO ABLOC FOR USE
C IN SUBROUTINE FUNP :
C
          IF (ICT .NE. 1) ABLOC(IC,IR) = ABLC
```

```
3 022.          C
3 023.    36       50     FINCR = FINCR + ABLC * V(IC)
3 024.          C
3 025.    37       60     FS = FS + FINCR
3 026.    38       70   CONTINUE
3 027.          C
3 028.    39             F(IR) = FS
3 029.    40     80 CONTINUE
3 030.          C
3 031.    41         RETURN
3 032.          C
3 033.          C --- JACOBIAN EVALUATION ----------------
3 034.          C
3 035.    42     90 REWIND 17
3 036.          C
3 037.          C ID-LOOP ITERATES THROUGH PARTIAL DERIVATIVES :
3 038.          C
3 039.    43         DO 130 ID = 1, N
3 040.          C
3 041.          C ROW-LOOP IR :
3 042.          C
3 043.    44           DO 130 IR = 1, N
3 044.    45             DFS = 0.D0
3 045.          C
3 046.          C COLUMN-LOOP IC :
3 047.          C
3 048.    46             DO 120 IC = 1, NN
3 049.          C
3 050.          C EVALUATE CONTRIBUTIONS FROM NONLIN. MATRIX :
3 051.          C
3 052.    47               DINCR = 0.D0
3 053.    48               I1 = IC1(IC)
3 054.    49               I2 = IC2(IC)
3 055.    50               IF (I1 .NE. ID .AND. I2 .NE. ID) GO TO 100
3 056.    51               IF (I1 .EQ. ID .AND. I2 .EQ. ID) DINCR = ANL1(IC,IR) * V(I1)
3 057.               1      * 2
3 058.    52               IF (I1 .EQ. ID .AND. I2 .NE. ID) DINCR = ANL1(IC,IR) * V(I2)
3 059.    53               IF (I1 .NE. ID .AND. I2 .EQ. ID) DINCR = ANL1(IC,IR) * V(I1)
3 060.    54    100        IF (IC .NE. ID .OR. IC .GT. N) GO TO 120
3 061.          C
3 062.          C EVALUATE CONTRIBUTIONS FROM LIN. MATRIX :
3 063.          C
3 064.    55               READ (17) ACIT
3 065.    56               ABLT = 0.D0
3 066.    57               DO 110 ICF = 1, ICR
3 067.    58                 ACMB = ACIT(ICF)
3 068.    59                 ABLT = ABLT + CFN(ICF) * ACMB
3 069.    60    110        CONTINUE
3 070.          C
3 071.          C COPY CRIT. VALUE OF LIN. MATRIX INTO ABLOT FOR USE
```

```
        C IN SUBROUTINE FUNP :
        C
 61               IF (ICT .NE. 1) ABLOT(IC,IR) = ABLT
        C
 62               DINCR = DINCR + ABLT
        C
 63       120     DFS = DFS + DINCR
        C
 64       130 DF(IR,ID) = DFS
        C
 65           RETURN
        CDS   DEBUG SUBCHK
 66           END
```

```
         C
         C===========================================================
         C
       1      SUBROUTINE FUNP(V, N, ANU, F, DF, IND)
         C
         C FUNCTION: ASSEMBLE ODE-SYSTEM REPRESENTING THE DISCRETIZ-
         C           ATION OF THE NONAUTONOMOUS EQUATION FOR
         C           2ND (IP=1) OR 3RD (IP=2) ORDER
         C           PERTURBATIONS ON THE NS-EQUATION. ITS
         C           NONAUTONOMY IS INTRODUCED BY AN ADVECTIVE TERM
         C           WITH AN ANALYTIC APPROXIMATION FOR A TIME-
         C           DEPENDENT 1ST OR 2ND ORDER PERTURBATION ,WHICH ALSO
         C           INTRODUCES ITS FUNCTIONAL DEPENDENCE ON THE
         C           BIFURCATION PARAMETER ANU.
         C
         C SYMBOLS USED: N     = DIMENSION OF ODE-SYSTEM (I.E.12)
         C               NN    = COLUMN NO. OF ITS NONLIN. MATRIX (I.E.78)
         C               V.    = ARRAY HOLDING 3 SETS OF PERTURBATION
         C                       VELOCITIES OF THE 3 BIFURCATIONS
         C               F     = LHS OF ODE-SYSTEM TO BE EVALUATED
         C               DF    = JACOBIAN OF ODE-SYSTEM TO BE EVALUATED
         C               IND   = SWITCH CONTROLLING EVALUATION OF ODE-
         C                       SYSTEM OR ITS JACOBIAN (USED WITHIN
         C                       BIFOR2 ONLY)
         C              ANL1   = NONLIN. MATRIX OF SYSTEM
         C             ABLOC   = LIN. MATRIX OF SYSTEM AT CRIT. VALUE
         C                       OF BIFURCATION PARAMETER ANU
         C             ABLOT   = TRANSPOSE OF ABLOC
         C               A3D   = ARRAY HOLDING 1 ELEMENT FROM EACH OF
         C                       THE 12 ADDITIONAL MATRICES FOR THE
         C                       ADVECTIVE TERMS STEMMING FROM PERTURB-
         C                       ATION VELOCITIES
         C               A3T   = SINGLE PRECISION ARRAY HOLDING ELEMENTS
         C                       TRANSPOSE TO THE ONES IN A3D
         C               V1    = COMPLEX EIGENVECTOR OF THE JACOBIAN
         C                       MATRIX AT THE CRIT. VALUE OF THE BIFURC-
         C                       ATION PARAMETER ANU - USED IN EVALUATION
         C                       OF ANALYTIC APPROXIMATIONS FOR PERTURB-
         C                       ATION VELOCITIES (I.E. HOPF ORBITS) VP.
         C                       (OUTPUT OF BIFOR2)
         C                  VP = ARRAY HOLDING PERTURBATION VELOCITIES
         C                 PAR = PARAMETERS FOR EVALUATION OF ANALYT.
         C                       APPROXIMATION OF PERTURBATION VELOCITIES
         C                       (I.E. HOPF ORBITS) VP. (OUTPUT OF BIFOR2)
         C               ANU   = BIFURCATION PARAMETER - ON ENTRY: INIT.
         C                       GUESS - ON RETURN: CRIT. VALUE
         C               VS    = TECH. ARRAY FOR TRANSFER OF 2ND PERTUB-
         C                       ATION VELOCITY
         C          IC1 ,IC2 = INDEX CONTROL ARRAYS TO CORRELATE
         C                       PRODUCTS OF ELEMENTS OF V WITH THE
```

```
     C                         CORRECT COLUMN NO. OF ANL1
     C                IP     = INDICATOR FOR 2ND (=1) OR 3RD (=2)
     C                         BIFURCATION
     C
     C LOG.UNITS ADDRESSED: 15 : INPUT  - A3D
     C                      19 : WORK   - A3T
     C
  2        IMPLICIT REAL*8(A - H,O - Z)
  3        INTEGER N
  4        REAL*8 V(N), F(N), DF(N,N)
  5        COMPLEX*16 V1(12,3), CDEXP, DCMPLX, EZ
  6        REAL*4 A3D(12), A3T(12)
  7        COMMON /CBL/ ANL1(78,12), CF(13), ABLOC(12,12), ABLDT(12,12), V1,
          1       VP(12), PAR(10,3), ANUP(3), CP(2), CC(10), IX(10), VS(12),
          2       IC1(78), IC2(78), IS(2), IT(2), NT(2), NN, ICR, IP
  8        COMMON /CT/ ICT
     C
     C-----------------------------------------------------------
     C EVALUATE ANALYT. APPROXIMATION OF 1ST (IP=1) OR 2ND (IP=2)
     C PERTURBATION :
     C
  9        ANUPIP = ANUP(IP)
 10        PAR1IP = PAR(1,IP)
 11        D = ANU - ANUPIP
 12        EPS = D / PAR1IP
     C
     C IF RADICAND EPS BECOMES NEGATIVE DURING SECANT ITERATION
     C (SEARCH FOR CRIT. VALUE OF ANU (ICT=1)) ,SWITCH TO MIRROR
     C IMAGE W.R.T APEX ANUPIP OF ASSOC. PARABOLA REPS = DSQRT(EPS) :
     C
 13        IF (ICT .EQ. 1 .AND. EPS .LT. O) ANU = ANU - 2 * D
 14        REPS = DSQRT(DABS((ANU - ANUPIP)/PAR1IP))
     C
     C USER-OPTION FOR TIMES ON ANALYT. APPROXIMATIONS FOR PERTURB-
     C ATIONS SET BY ITERATIONS IN TIME-LOOPS (IS(IP)=O) OR BY USER
     C (IS(IP)=1 ,NT(IP)=TIME) :
     C
 15        IF (IS(IP) .EQ. O) EZ = CDEXP(DCMPLX(O.DO,6.283185307179586476900*
          1IT(IP)/NT(IP)))
 16        IF (IS(IP) .EQ. 1) EZ = CDEXP(DCMPLX(O.DO,6.283185307179586476900*
          1CP(IP)))
     C
     C CALCULATE 1ST AND/OR 2ND PERTUBATIONS AT THEIR CRIT. VALUES :
     C
 17        DO 10 IV = 1, N
 18          VPP = REPS * DREAL(EZ*V1(IV,IP))
     C
     C COPY CRIT. VALUE OF 1ST PERTURBATION INTO VS :
     C
 19          IF (IP .EQ. 1 .AND. ICT .EQ. O) VS(IV) = VPP
```

```
C
C COPY 1ST PERTURBATION INTO VP :
C
        IF (IP .EQ. 1) VP(IV) = VPP
C
C COPY 1ST (CRIT. VALUE) + 2ND PERTURBATION INTO VP :
C
        IF (IP .EQ. 2) VP(IV) = VPP + VS(IV)
   10 CONTINUE
C
C-------------------------------------------------------------------
C
      IF (IND .EQ. 1) GO TO 80
C
C --- FUNCTION EVALUATION -------------------------------------------
C
      REWIND 15
C
C ROW-LOOP IR :
C
      DO 70 IR = 1, N
C
C COLUMN-LOOP IC :
C
        FS = 0.D0
        DO 60 IC = 1, NN
C
C EVALUATE NONLIN. MATRIX :
C
          I1 = IC1(IC)
          I2 = IC2(IC)
          FINCR = ANL1(IC,IR) * V(I1) * V(I2)
          IF (IC .GT. N) GO TO 50
C
C ASSEMBLE ADVECT. TERMS STEMMING FROM PERTURBATIONS :
C
   20     READ (15) A3D
          CL = 0.D0
          DO 30 IV = 1, N
            A3DP = A3D(IV)
            CL = CL + A3DP * VP(IV)
   30     CONTINUE
C
C ADD CRIT. VALUE OF ADVECTIVE MATRIX ABLOC OF MEAN FLOW :
C
   40     FINCR = FINCR + (ABLOC(IC,IR) - CL) * V(IC)
C
   50     FS = FS + FINCR
   60   CONTINUE
C
```

```
         F(IR) = FS
   70 CONTINUE
C
      RETURN
C
C --- JACOBIAN EVALUATION -----------------
C
   80 REWIND 19
C
C ID-LOOP ITERATES THROUGH PARTIAL DERIVATIVES :
C
      DO 120 ID = 1, N
C
C ROW-LOOP IR :
C
        DO 120 IR = 1, N
          DFS = 0.D0
C
C COLUMN-LOOP IC :
C
          DO 110 IC = 1, NN
C
            DINCR = 0.D0
            I1 = IC1(IC)
            I2 = IC2(IC)
            IF (I1 .NE. ID .AND. I2 .NE. ID) GO TO 90
            IF (I1 .EQ. ID .AND. I2 .EQ. ID) DINCR = ANL1(IC,IR) * V(I1)
     1       * 2
            IF (I1 .EQ. ID .AND. I2 .NE. ID) DINCR = ANL1(IC,IR) * V(I2)
            IF (I1 .NE. ID .AND. I2 .EQ. ID) DINCR = ANL1(IC,IR) * V(I1)
   90       IF (IC .NE. ID .OR. IC .GT. N) GO TO 110
C
C ASSEMBLE ADVECT. TERMS STEMMING FROM PERTURBATIONS :
C
            READ (19) A3T
            CL = 0.D0
            DO 100 IV = 1, N
              A3TP = A3T(IV)
              CL = CL + A3TP * VP(IV)
  100       CONTINUE
C
C ADD CRIT. VALUE OF ADVECTIVE MATRIX ABLOC OF MEAN FLOW :
C
            DINCR = DINCR + ABLOT(IC,IR) - CL
C
  110     DFS = DFS + DINCR
  120 DF(IR,ID) = DFS
C
       RETURN
CDS    DEBUG SUBCHK
```

3 285. 67 END

```
        C
        C=======================================================
        C
 1            SUBROUTINE MFC
        C
        C FUNCTION: EVALUATE MEAN FLOW AT CRIT. POINT OF 1ST BIFURCATION.
        C
        C SYMBOLS USED: I ,J ,K  = INT. POWERS OF X ,Y ,Z-DEPENDENCIES
        C                          OF MEAN FLOW COMPONENTS
        C           XI ,YI ,ZI = CARTES. COORDINATES OF POINTS
        C                          SELECTED BY DISCRETIZATION
        C                     VC = ARRAY CONTAINING MEAN FLOW AT
        C                          CRIT. POINT
        C                    CFT = PHYS. FLOW (MEAN) PARAMETERS AT
        C                          CRIT. POINT
        C                     C  = PARAMETERS SELECTED FROM CFT
        C                          USED IN CURRENT MEAN FLOW
        C
        C LOG.UNITS ADDRESSED:  1 : OUTPUT - VC
        C                       4 : INPUT  - CONTROL PARAMETERS
        C                       6 : OUTPUT - ERROR MESSAGES - TERMINAL
        C                      14 : INPUT  - XI ,YI ,ZI
        C
 2            LOGICAL*1 LFMT(1) /'*'/
 3            REAL*8 X, Y, Z, VC(12), CFT(13), C(13) /13*0.D0/
 4            REAL*4 XI, YI, ZI
 5            INTEGER I(3), J(3), K(3), NCF /13/
 6            COMMON /LF/ LCF(13)
 7            COMMON /CBT/ CFT
        C
        C READ POWERS FROM CONTROL PARAMETER FILE (LOG.UNIT 4) :
        C
 8            READ (4'143000,LFMT) I
 9            READ (4'144000,LFMT) J
10            READ (4'145000,LFMT) K
        C
        C SELECT COEFFS. USED IN CURRENT MEAN FLOW :
        C
11            ICR = 0
12            DO 10 ICF = 1, NCF
13              IF (LCF(ICF) .EQ. 0) GO TO 10
14              ICR = ICR + 1
15              C(ICF) = CFT(ICR)
16         10 CONTINUE
        C
        C EVALUATE MEAN FLOW AT CRIT. POINT :
        C
17            DO 20 II = 3, 12, 3
18              READ (14) XI, YI, ZI
19              X = XI
```

```
3 336.      20          Y = YI
3 337.      21          Z = ZI
3 338.      22          VC(II - 2) = C(1) + C(5) * X ** I(1) + C(6) * Y ** J(1) + C(7) *
3 339.               1   Z ** K(1)
3 340.      23          VC(II - 1) = C(2) + C(8) * X ** I(2) + C(9) * Y ** J(2) + C(10)
3 341.               1  * Z ** K(2)
3 342.      24          VC(II) = C(3) + C(11) * X ** I(3) + C(12) * Y ** J(3) + C(13) *
3 343.               1  Z ** K(3)
3 344.            C
3 345.            C PRINT OUT FLOW AT SELECTED POINTS AND WRITE INTO LOG.UNIT 1 :
3 346.            C
3 347.      25          WRITE (6,30) X, Y, Z
3 348.      26          WRITE (6,30) VC(II - 2), VC(II - 1), VC(II)
3 349.      27       20 CONTINUE
3 350.      28          WRITE (1) VC
3 351.            C
3 352.      29       30 FORMAT (3G15.7)
3 353.      30          RETURN
3 354.            CDS   DEBUG SUBCHK
3 355.      31          END
```

```
C     PROGRAM ODE.M
C
C FUNCTION: SOLVE THE NONAUTONOMOUS SYSTEM OF THE DISCRETIZED
C           NS-EQ. FOR THE 2ND PERTUBATIONS (V) BY RUNGE-KUTTA.
C           THE 1ST PERTURBATIONS (VS) ENTER THE SYSTEM AS AN
C           ANALYTIC APPROXIMATION AND THEREBY INDUCE ITS NON
C           AUTONOMY.
C           INDUCE FOR IRP.LT.K RANDOM PERTURBATIONS AT TIMESTEP K
C           AT 4 POINTS WITH 3 VELOCITY COMPONENTS BY DISPLACING
C           THEM IN THEIR 4 3 DIM. VELOCITY SPACES BY VECTORS DET
C           ERMINED IN LENGTH AND DIRECTION BY GAUSSIAN AND HOM
C           OGENEOUS RANDOM NUMBER GENERATORS. THE AMOUNT OF
C           TIMESTEPS WITH UNPERTURBED VELOCITIES IS DETERMINED
C           BY A GAUSSIAN RANDOM GENERATOR AS WELL.
C
C SUBROUTINES USED: RKF45: 4TH-5TH ORDER RUNGE-KUTTA-FEHLBERG
C                          SOLVER (MATH:MATHLIB)
C                   F45:   ASSEMBLES NONAUTONOMOUS NS-SYSTEM
C                   GGNQF: GAUSSIAN RANDOM NUMBER GENERATOR
C                   GGSPH: HOMOGENEOUS RANDOM NUMBER GENERATOR
C                          OVER UNIT SPHERE
C                          (BOTH IMSLLIB)
C
C LOG.UNITS ADRESSED: 1 - INPUT - FROM BIF3A (PARD,V1D,ANUCD)
C                     2 - OUTPUT- 2ND PERT. (T,V)
C                     3 - OUTPUT- 1ST PERT. (T,VS)
C                     4 - INPUT - CONTROL PARAMETERS
C                     6 - OUTPUT - ERROR MESSAGES - TERMINAL
C                     7 - INPUT - IC1 FROM DSCRB
C                     8 - INPUT - IC2 FROM DSCRB
C                    12 - INPUT - ANL1 FROM DSCRB
C                    13 - INPUT - ABLOC FROM BIF3A
C                    15 - INPUT - A3D FROM ADV
C
C SYMBOLS USED: K      = COUNT OF TIMESTEPS
C               MK     = MAX.NO. OF TIMESTEPS
C               N      = DIM. OF ODE SYSTEM (I.E. 12 HERE)
C               NN     = NO. OF COLUMNS IN NONLIN. MATRIX
C               IRD    = NO. OF UNPERTURBED TIMESTEPS PER POINT
C               ICD    = COUNT OF UNPERTURBED TIMESTEPS
C               ABLOCD = LINEAR MATRIX OF SYSTEM
C               ANL1   = NONLIN.MATRIX OF SYSTEM
C               A3D    = ADVECTIVE MATRIX OF SYSTEM
C               IRP    = SWITCH FOR RANDOM PERTURBATIONS
C               T      = CURRENT TIME IN ODE SOLVER
C               TS     = TIMESTEP IN ODE SOLVER
C               TEND   = TIME FOR WHICH THE SYSEM IS SOLVED
C               ANUC   = CRIT. VALUE OF BIFURCATION PARAMETER
C                        FOR FIRST BIFURCATION
C               ANUCD  = ANUC IN DOUBLE PRECISION
```

```
C               PAR    = PARAMETERS FOR ANALYT. APPROXIMATION
C                        OF FIRST PERTURBATION
C               PARD   = PAR IN DOUBLE PRECISION
C               V1     = COMPLEX EIGENVECTOR OF SYSTEM AT ANUC
C                        -  USED TO CALCULATE VS
C               V1D    = V1 IN DOUBLE PRECISION
C               VS     = ANALYT. APPROXIMATION OF FIRST
C                        PERTURBATION
C               V      = SECOND PERTURBATION - I.E. UNKNOWN
C                        WHICH THE SYSTEM IS SOLVED FOR
C               ANUP   = CURRENT SETTING OF BIFURCATION
C                        PARAMETER IN VS
C             IC1 ,IC2 = INDEX CONTROL ARRAYS TO CORRELATE
C                        PRODUCTS OF ELEMENTS OF V WITH THE
C                        CORRECT COLUMN NO. OF ANL1
C               VM     = V AS 3*4 ARRAY
C               SS     = VARIANCE OF GAUSSIAN RANDOM DISTRIB
C                        UTION FOR SIZE OF RANDOM PERTURB
C                        ATIONS IN SPACE
C               ST     = VARIANCE OF GAUSSIAN RANDOM DISTRIB
C                        UTIONS FOR RANDOM NUMBER OF UNPERTUB
C                        ED TMESTEPS
C               Z      = DIRECTION OF RANDOM PERTURBATIONS -
C                        I.E. OUTPUT OF GGSPH
C               RDF    = REDUCTION FACTOR FOR UNBOUNDED GROWTH
C               RELERR ,ABSERR ,IFLAG ,WORK ,IWORK :
C                        TECH. PARAMETERS AND ERROR CODE
C                        FOR RKF45
C               DSEED ,NRD ,IOPT ,IZ ,IERR :
C                        TECH. PARAMETERS AND ERROR CODE
C                        FOR GGNQF ,GGSPH
C
      LOGICAL*1 LFMT(1) /'*'/
      INTEGER K, MK, N, IWORK(5), IZ /1/, IRD(4) /4*0/, ICD(4) /4*0/,
     1        NRD /1/, IOPT /3/, IERR
      REAL V(12), VM(3,4), ANUP(3), WORK(75)
      REAL Z(1,3), GGNQF, SF, SS, ST
      REAL T, TS, TEND
      REAL*8 ANUCD, PARD(10), ABLOCD(12,12), ABLOTD(12,12), DSEED
      COMPLEX*16 V1D(12)
      COMPLEX V1(12)
      EQUIVALENCE (V(1),VM(1,1))
C
      COMMON /CBK/ V1, PAR(10), IC1(78), IC2(78), REPS, VS(12), PR
      COMMON /MTRX/ ABLOC(12,12), ANL1(78,12), A3D(12,144)
      COMMON /NEQAT/ N, NN
C
      EXTERNAL F45
C
      NN = 78
      N = 12
C
C READ CONTROL PARAMETERS AND SWITCHES FROM LOG. UNIT 4 :
C
      READ (4'10000,LFMT) ANUP
      READ (4'60000,LFMT) IND, IND2, IFLAG, IFLAG2, IFLAG3, IFLAGR, M34
      READ (4'62000,LFMT) MK, IRW
      READ (4'64000,LFMT) T, IRP, SS, ST, DSEED
      READ (4'66000,LFMT) TS, RDF, FELMX
      READ (4'68000,LFMT) TOL, H, RELERR, ABSERR, RFCTR, AFCTR
      READ (4'70000,LFMT) V
C
C PARAMETERS FOR ANALYT.APPROXIMATION OF 1ST PERTURBATION
C AND LINEAR MATRIX ABLOCD FOR ODE-SYSTEM ARE READ FROM
C LOG.UNITS 1 AND 13 AND CONVERTED TO SINGLE PRECISION.
C NONLINEAR MATRIX ANL1 ,ADVECTIVE MATRIX A3D AND INDEX
C CONTROL ARRAYS FOR NONLIN.MATRIX IC1 ,IC2 ARE READ FROM
C LOG.UNITS 12 ,15 , 7 ,8. (ABLOTD UNUSED)
C
      READ (1) ANUCD, (PARD(JP1),JP1=1,10), (V1D(JV1),JV1=1,N)
      READ (13) ABLOCD, ABLOTD
      DO 10 JR = 1, N
   10 READ (12) (ANL1(JC,JR),JC=1,NN)
      NTN = N * N
      DO 20 KR = 1, NTN
   20 READ (15) (A3D(KC,KR),KC=1,N)
C
      READ (7) IC1
      READ (8) IC2
      ANUC = ANUCD
      DO 40 IR = 1, N
        IF (IR .LE. 10) PAR(IR) = PARD(IR)
        V1(IR) = V1D(IR)
        DO 30 IC = 1, N
          ABLOC(IC,IR) = ABLOCD(IC,IR)
CT        ABLOT(IC,IR) = ABLOTD(IC,IR)
   30   CONTINUE
   40 CONTINUE
C
C CALCULATE PARAMETERS FOR ANALYT. APPROXIMATION OF 1ST
C PERTURBATION VS :
C
      EPS = (ANUP(2) - ANUC) / PAR(1)
      PR = (1. + PAR(2)*EPS) / PAR(8)
      P = PR * 6.28318530717958647690O
      REPS = SQRT(EPS)
C
C WRITE TMESTEP TS ,REDUCTION FACTOR RDF AND PERIOD P
C ON LOG.UNIT 2 :
C
```

```
 44          WRITE (2) TS, RDF, P
       C
       C-------- TIME LOOP FOR RKF45 AND RANDOM PERTURBATIONS -----
       C
 45          TEND = T
 46          DO 130 K = 1, MK
 47            TEND = FLOAT(K) * TS
       C
       C SKIP RKF45 IF PURE RANDOM WALK IS TO BE PERFORMED (USER-OPTION) :
       C
 48          IF (IRW .EQ. 1) GO TO 90
       C---------------------------------------------------------------
 49          K34 = 0
 50     50   CALL RKF45(F45, N, V, T, TEND, RELERR, ABSERR, IFLAG, WORK,
          1       IWORK)
       C
       C--------- SECTION FOR ERROR TERMINATION CONTROL -----------
       C
 51          K34 = K34 + 1
 52          IF (IFLAG .LE. 2) GO TO 80
 53          WRITE (5,160) K, T, IFLAG
 54          WRITE (5,230) (V(II),II=1,N)
 55          IF (IFLAG .NE. 7) GO TO 60
 56          IFLAG = IFLAG3
 57          GO TO 80
 58     60   IF (IFLAG .LE. 4 .AND. K34 .LT. M34) GO TO 50
 59          IF (IFLAG .NE. 6) GO TO 140
 60          IFLAG = IFLAG2
       C
       C FOR UNBOUNDED GROWTH (IFLAG=6) ,THE LAST SUCCESSFUL
       C OUTPUT V IS REDUCED BY MULTIPLICATION WITH RDF.
       C
 61          DO 70 M = 1, 12
 62            V(M) = V(M) * RDF
 63     70   CONTINUE
       C-----------------------------------------------------------------
       C SKIP RANDOM PERTURBATIONS (USER-OPTION) :
       C
 64     80   IF (K .LT. IRP) GO TO 120
       C
       C------- GAUSSIAN RANDOM PERTURBATIONS IN SPACE & TIME --------
       C
 65     90   DO 110 JT = 1, 4
 66             IF (IRD(JT) .NE. ICD(JT)) GO TO 100
 67             IFLAG = IFLAGR
 68             CALL GGSPH(DSEED, NRD, IOPT, IZ, Z, IERR)
 69             SF = SS * GGNQF(DSEED)
 70             VM(1,JT) = VM(1,JT) + Z(1,1) * SF
 71             VM(2,JT) = VM(2,JT) + Z(1,2) * SF
 72             VM(3,JT) = VM(3,JT) + Z(1,3) * SF
```

```
      C
                IRD(JT) = IFIX(ABS(ST*GGNQF(DSEED)))
                WRITE (6,210) JT, (Z(1,I),I=1,3), SF, IRD(JT)
                ICD(JT) = -1
        100     ICD(JT) = ICD(JT) + 1
        110   CONTINUE
      C------------------------------------------------------------------
      C 1ST (VS) AND 2ND (V) PERTURBATIONS ARE WRITTEN INTO
      C LOG.UNITS (OUTPUT) 3 AND 2 TOGETHER WITH TIME T.
      C
        120   WRITE (2) T, V
              WRITE (3) T, VS
      C
        130 CONTINUE
      C
      C------------ END OF TIME LOOP -------------------------------
      C
      C OUTPUT OF LAST TIME STEP V AND ITS EUCLIDEAN NORM ENORM1
      C (ENORM2 UNUSED PRESENTLY). GENERATION OF ERROR MESSAGES
      C AND THEIR OUTPUT WITH ALL RELEVANT PARAMETERS.
      C
        140 CONTINUE
      C
            SQNRM1 = 0.
            SQNRM2 = 0.
            DO 150 I = 1, N
              VI = V(I)
              SQNRM1 = SQNRM1 + VI * VI
              WRITE (6,170) I, VI
        150 CONTINUE
      C
            ENORM1 = SQRT(SQNRM1)
            ENORM2 = SQRT(SQNRM2)
            WRITE (6,200) K, ENORM1, ENORM2, P, T, TEND
            WRITE (6,220) RELERR, ABSERR, IFLAG
      C
        160 FORMAT ('K=', I5, '  T=', F8.4, '  IFLAG=', I3)
        170 FORMAT (I3, 3X, 3X, G10.2)
        180 FORMAT (I3, 3X, 3(3X,G10.2))
        190 FORMAT ('K=', I5, 3X, 'ENORM1 = ', G15.5, 3X, 'ENORM2 = ', G15.5)
        200 FORMAT ('K=', I5, 3X, 'ENORM1 = ', G15.5, 3X, 'ENORM2 = ', G15.5,
           1       ' P=', F8.4, ' T=', F8.4, ' TEND=', F8.4)
        210 FORMAT (I1, 3X, 4G15.7, 3X, I3)
        220 FORMAT ('RELERR=', G15.7, '  ABSERR=', G15.7, '  IFLAG=', I3)
        230 FORMAT (12(G10.2,1X))
        240 FORMAT (13(2X,I1,2X))
        250 FORMAT (13F5.2)
        260 FORMAT (1X, I1, 2X, I2, 2X, G10.2, 2(4X,G10.2)/8(6(1X,G10.2)/))
        270 FORMAT (13F8.3)
      C
```

```
105          STOP
      CDS    DEBUG SUBCHK
106          END
```

```
  1         SUBROUTINE F45(T, V, F)
       C
       C FUNCTION: ASSEMBLE ODE-SYSTEM REPRESENTING THE DISCRETIZ-
       C           ATION OF THE NONAUTONOMOUS EQUATION FOR 2ND
       C           ORDER PERTURBATIONS ON THE NS-EQUATION. ITS
       C           NONAUTONOMY IS INTRODUCED BY AN ADVECTIVE TERM
       C           WITH AN ANALYTIC APPROXIMATION FOR A TIME-
       C           DEPENDENT 1ST ORDER  PERTURBATION.
       C
       C LOG.UNITS ADRESSED: 6 - OUTPUT - TERMINAL (TIME T)
       C
       C SYMBOLS USED: N    = DIMENSION OF ODE-SYSTEM
       C               NN   = COLUMN NO. OF ITS NONLIN. MATRIX
       C               V    = VECTOR OF UNKNOWN 2ND ORDER PERTURB-
       C                      ATION VELOCITIES
       C               F    = RHS-VECTOR OF ODE-SYSTEM
       C               V1   = COMPLEX EIGENVECTOR OF ODE-SYSTEM RE-
       C                      PRESENTING THE DISCRETIZATION OF THE
       C                      EQUATION FOR THE 1ST ORDER PERTURBATION
       C                      (USED WITHIN THE ANALYTIC APPROXIMATION
       C                      FOR THE 1ST ORDER PERTURBATION)
       C               VS   = ANALYTIC APPROXIMATION OF 1ST ODER PER-
       C                      TURBATION
       C                T   = TIME
       C               A3D  = ARRAY CONTAINING ADVECTIVE MATRICES IN
       C                      THE SEQUENCE THEY (I.E. THEIR ELEMENTS)
       C                      ENTER ROW- AND COLUMN LOOPS
       C             ABLOC  = LIN.MATRIX
       C              ANL1  = NONLIN. MATRIX
       C          IC1 ,IC2  = INDEX CONTROL ARRAYS FOR VELOCITY PRO-
       C                      DUCT VECTOR TO BE MULTIPLIED WITH NON-
       C                      LIN. MATRIX
       C
  2         INTEGER N
  3         REAL V(12), F(12), T
  4         COMPLEX V1(12), CEXP, CMPLX, EZ
  5         COMMON /CBK/ V1, PAR(10), IC1(78), IC2(78), REPS, VS(12), PR
  6         COMMON /MTRX/ ABLOC(12,12), ANL1(78,12), A3D(12,144)
  7         COMMON /NEQAT/ N, NN
       C
       C-----------------------------------------------------------
       C EVALUATE ANALYT. APPROXIMATION OF PERTURBATION
       C VELOCITY VS :
       C
  8         EZ = CEXP(CMPLX(0.,T/PR))
       C
  9         DO 10 IV = 1, N
 10            VS(IV) = REPS * REAL(EZ*V1(IV))
 11      10 CONTINUE
       C
```

```
C----- FUNCTION EVALUATION ---------------------------------
C
C INIT. ROW/COLUMN INDEX FOR A3D :
C
      KA3D = 0
C
C ROW-LOOP IR :
C
      DO 40 IR = 1, N
        FS = 0.
C
C COLUMN-LOOP IC :
C
        DO 30 IC = 1, NN
C
C EVALUATE NONLIN. MATRIX :
C
          I1 = IC1(IC)
          I2 = IC2(IC)
          FS = FS - ANL1(IC,IR) * V(I1) * V(I2)
          IF (IC .GT. N) GO TO 30
C
C ASSEMBLE ADVECTIVE MATRIX STEMMING FROM PERTURBATION :
C
          KA3D = KA3D + 1
          CL = 0.
          DO 20 IV = 1, N
            CL = CL + A3D(IV,KA3D) * VS(IV)
   20     CONTINUE
C
C ADD LIN. (ADVECTIVE) MATRIX ABLOC STEMMING FROM MEAN FLOW :
C
          FS = FS + (ABLOC(IC,IR) - CL) * V(IC)
C
   30   CONTINUE
C
        F(IR) = FS
   40 CONTINUE
C
C MONITOR TIME T :
C
      WRITE (6,50) T
   50 FORMAT (' T=', G15.6)
C
      RETURN
CDS   DEBUG SUBCHK
      END
```

```
     FILE 4.NS.A: CONTROL PARAMETERS / PHYS.PARAMETERS FOR BIF3A ---
     CX(1)     CY(2)       CZ(3)      VSC(4)     CXX(5)   CXY(6)     CXZ(7)     CYX(8)    CYY(9)    CYZ(10)    CZX(11)
CZY(12)   CZZ(13)
        1         1           1          1          0        1          0          0         0         1          1
  0         0
       -9.       -6.         -1.        -1.        .0       9.         .0         .0        .0        6.         1.
 .0         .0
     CONSTANTS FOR CRIT.PARAMETER FCTNS. - CC(10) :
     0.   0.    0.    0.    4.          5.   3.   0.   0.   0.
     EXPONENTS FOR CRIT.PARAMETER FCTNS. - IX(10) :
     0    0     0     0     0           0    0    0    0    0
     INIT.GUESSES FOR BIF.PARAMETERS  -  ANUP(3) :
     .2423356     .32       .00
     SWITCHES FOR SEQUENTIAL INCREMENTATION OF ANUP(2),(3) - IA12, IA23 :
     0    0
     SEQUENTIAL INCREMENTATION OF ANUP(2),(3) - DA12, DA23 :
     .1    .1
     INIT.GUESSES FOR VELOCITIES VI1/X/Y/Z,VI2/X/Y/Z,VI3/X/Y/Z - VI(3,3):
     00.0 00.0 00.0, .0  .0  .0 , .0  .0  .0
     ___________________________________________________________________
     **    ORBIT CONTROL PARAMETERS AND SWITCHES FOR BIF3A
     TIME SETTINGS ON ORBIT 1 AND 2  -  CP(2) :
           .0        .5
     NO. OF STEPS IN TIME LOOPS  -  NT(2) :
     1  1
     INITIAL TIME SWITCHES FOR INCREMENT VERSION  -  IS(2) :
     1  1
     NO. OF ORBIT SEQUENCES  -  NO :
     1
     SWITCHES FOR ANUP(1),ANUP(2),ANUP(3)  -  IR1, IR2, IR3 :
     0   0   0
     SKIP SWITCHES FOR THE FIRST TWO BIFURCATIONS  -  IG1, IG2 :
     0  0
     INITIAL TIME SETTINGS IN TIME LOOPS  -  TI1, TI2 :
     0. 0.
     ___________________________________________________________________
     **    TECH.PARAMETERS AND SWITCHES FOR BIFOR2
     U1, U2, U3 :
     1.0D-15  1.0D-15  1.0D-15
     EPSR1, EPSR2, EPSR3 :
     1.0D-4  1.0D-4  1.0D-4
     EPS1, EPS2, EPS3 :
     1.0D-15  1.0D-15  1.0D-15
     NSIG1, NSIG2, NSIG3 :
     13  13  13
```

C6

```
ITMAX:
20
ICK(3):
0 0 0
MTH1, MTH2, MTH3 :
1  1  1
JJOB1, JJOB2, JJOB3 :
1  1  1
IPRNT1, IPRNT2, IPRNT3 :
2  2  2
_____________________________________________________
** TECH.PARAMETERS FOR O.D.E.-SOLVERS
UNUSED! SWITCH; DVERK (LVG=0) ¬ DGEAR (LVG=1) ¬ RKF45 (LVG=2) -  LVG
2
UNUSED! INDEX(DGEAR)
1
IND,IND2(DVERK- UNUSED),IFLAG,IFLAG2,IFLAG3,IFLAGR,M34(RKF45)
2   2                    1      1      1      1      4
NO.OF TIMESTEPS ,RANDOM WALK OPTION  - MK ,IRW :
                                      100   0
INIT.TIME ,RAND.PERT.SWITCH ,VAR. - T ,IRP  ,SS  ,ST  ,DSEED:
                                   .0  110  .05  1.5  923456.0D0
STEPSIZE ,RED.FACTOR AND MAX.NO OF FCTN.EVLS.- TS      ,RDF  ,MAXNFE:
                                               .05    1.0E-12  1000
TOL(DGEAR,DVERK),H(DGEAR) ,RELERR ,ABSERR ,RFACTR ,AFACTR(ALL RKF45)
1.0E-3            1.0E-3   1.0E-10 1.0E-10   10       10
INITIAL V   -  V(12)
.1    .0    .0        .0     .0     .0        .0     .0     .0        .0     .0     .0
_______________________________________________________
** PARAMETERS FOR PLOTTING ROUTINE
NO. OF INPUT FILE - IFL1 ,IFL2 (IFL2=0 if only one file used):
2   3
SDI ,SDJ  ,PS  ,AD  ,INTVF ,FHTNO ,AR :
0.   0.   28.  .05    4     .01    4
XMX ,XMN ,YMX ,YMN  ,FRL ,FRR ,FRB ,FRT :
 2.5,-2.5, 2.5,-2.5 ,.12  .1   .12   1
MAX. NO. OF ITERATIONS - MXNT :
8200
NO. OF UNGRAPHED INIT. ITERATIONS - ITNG :
2
COORDS. TO BE GRAPHED (K01=0 : TIME ON ABSCISSA) - K01 ,K02 ,K03 :
1    2    2
TRACK MARK PARAMETERS - ITMK  ,FHT  ,MKA ,MKO  ,ANG  ,ISMBL :
                          2    .015   1    8    0.0   -1
COMBINATIONS  OF COORDINATES   -  KO(24) :
```

```
1 2  1 3  2 3  4 5  4 6  5 6  7 8  7 9  8 9  10 11  10 12  11 12
ANGLES FOR ROTATION IN 3D  -  XRAD ,YRAD:
O.  O.
SWITCHES AND POINCARE MAP POS.;  IFQ ,IPT(LE.1) ,IPC ,TP
                                  0    1          0   1.5

SWITCH,COORD.,LOCATION AND SLAB THICKNESS FOR CUTS - ISC,KOS,SC,ST:
O  3  2.  .1

PARAMETERS FOR FREQENCY ANALYSIS (FAST FOURIER TRANSFORM)
LOG.UNITS OF INPUT FILES - IFL1 ,IFL2 :
2  3
SELECTOR INDEX - .LE.O: 1 , .GT.O: 2   TIME SERIES  - IND :
2
SAMPLE SIZE ,READ INDEX - MXNT ,IRD :
8200    1
COORDS. TO BE TESTED - KO1 ,KO2 :
1   2
!!INOP!! L-PARAMETER (POWER OF 2) - L :
256
NO. OF FREQS./PERIODS,ETC. PRINTED OUT  -  MOP :
10
next 3 lines unused presently
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb

PARAMETERS FOR WAVENUMBER ANALYSIS (FAST FOURIER TRANSFORM)
LOG.UNITS OF INPUT FILES - IFL1 ,IFL2 :
2  3
SELECTOR INDEX - .LE.O: 1 , .GT.O: 2  SPACE SERIES  - IND :
2
SAMPLE SIZE ,READ INDEX (WNAS ONLY) - MXNT ,IRD :
8200     1
INDICES FOR FTFREQ - IND(3) ,IND(5) ,IND(6) ,XIND(2) :
O   O   1    O.
!!INOP!!SELECTOR FOR SPECTRUM TO BE PLOTTED (1 OR 2) - SOP :
1
NO. OF LINES PRINTED OUT ,NO. OF UNGRAPHED STEPS - MOP ,ITNG :
10    1
VIRT.WINDOW ,TITLE - SIZE ,IN INCHES - SVX ,SVY ,TSZ :
5.     3.5    .12
CGPL-PARAMETER   -  KC(3)  :
4   24    1
CGPL-PARAMETER FOR AXIS SCALES  -   KA ,KB  :
2  2

```

```
   PARAMETERS FOR DISCRETIZATION PROGRAM DSCRB
 1.DATA LINE: MAX.NO.OF INCREMENTS IN X-,Y-,Z- DIRECTION
    2   2   2
 2.-4.DATA LINE: DISCRETIZATION INCREMENTS IN X-,Y-,Z-DIRECTION
    000.000   001.000   010.000   003.000   004.000
    000.000   001.000   002.000   003.000   004.000
    000.000   001.000   006.000   003.000   004.000
 POWERS OF SPATIAL DEPENDENCIES FOR MEAN FLOW IN :
   X-COMPONENT:        Y-COMPONENT:       Z-COMPONENT:
      1                   3                  2
      2                   1                  3
      3                   2                  1
 EXPONENTS PRIOR TO 29.06.83 (FILE R.11)
      1                   3                  2
      2                   1                  3
      3                   2                  1
 EXP.FOR X-,Y-,Z- FLOW COMP.(must not exceed 3 or 6 rsp.)(F R.11.4)
      4                   4                  4
      4                   4                  4
      4                   4                  4
 EXPONENTS PRIOR TO 30.06.83 (FILE R.11.1)
      1                   2                  3
      1                   2                  3
      1                   2                  3
 EXPONENTS OF FILE R.11.2
      1                   2                  1
      3                   2                  3
      1                   2                  1
 EXPONENTS OF FILE R.11.3
      3                   3                  3
      3                   3                  3
      3                   3                  3
 EXPONENTS OF FILE  11.1
      1                   1                  1
      2                   2                  2
      3                   3                  3
 EXPONENTS PRIOR TO 29.06.83 (FILE 11)
      1                   1                  1
      2                   1                  1
      3                   2                  1
End of file
```